Naval Research Laboratory

Washington, DC 20375-5320

NRL/FR/5324—92-9397

High-resolution Array with Prony, MUSIC, and ESPRIT Algorithms

B. T. ROOT

Advanced Radar Branch
Radar Division

August 25, 1992

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 25, 1992	Final

4. TITLE AND SUBTITLE

High-resolution Array with Prony, MUSIC, and ESPRIT Algorithms

5. FUNDING NUMBERS

PE OMN 204577N
WU DN291-215
TA 148-NRL-92-VARI-002

6. AUTHOR(S)

B. T. Root

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Washington, DC 20375-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/FR/5324—92-9397

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Space and Naval Warfare Systems Command
Washington, DC 20363-5100

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report investigates the array high-resolution properties of three algorithms: the Prony algorithm, the MUSIC algorithm, and the ESPRIT algorithm. MUSIC has been much studied due to its excellent high-resolution properties in all but the most correlated cases. A variation of it called spatially averaged MUSIC performs well even in the case of perfectly correlated signals. The Prony technique has been less studied in an array processing context, but is shown to perform as well as spatially averaged MUSIC. ESPRIT is a recently developed eigenspace-based technique that has the same excellent resolution properties as MUSIC, but is much more computationally efficient. It can also be adapted to the correlated case.

14. SUBJECT TERMS

Array processing
High resolution

15. NUMBER OF PAGES

77

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std Z39-18
298-102

CONTENTS

HIGH-RESOLUTION ARRAY WITH PRONY, MUSIC, AND ESPRIT ALGORITHMS

1. INTRODUCTION

This report investigates the array high-resolution properties of three algorithms: Prony, MUSIC (MUltiple SIgnal Classification), and ESPRIT (Estimation of Signal Parameters via Rotational Invariance Techniques).

The Prony algorithm [1] for estimating the parameters of sinusoids in noise is very old.* (Its discoverer, the Baron de Prony, lived around the time of the French revolution.) Unlike MUSIC and most other eigenspace-based methods, it can work with correlated signals, but because of the assumed signal model (sinusoids in noise), it is *geometry-dependent.* Specifically, it must assume that the array is linear, and, therefore, the range of applicability is greatly reduced. Nevertheless, the importance of resolving correlated signals makes it an interesting high-resolution algorithm.

The MUSIC [2] algorithm has received a lot of attention due to its excellent behavior when the signals have even a small degree of decorrelation. However, it breaks down when the signals are perfectly or nearly perfectly correlated (in the statistical sense explained below). Unfortunately, this can occur in many cases of interest, especially with multipath, and it is then difficult to distinguish multipath from true targets.

ESPRIT [3] is a recently developed eigenspace-based technique that has the same excellent resolution properties as MUSIC, but is much more computationally efficient. It can also be adapted to the correlated case.

This report concentrates primarily on an investigation of the Prony algorithm in the case of perfect correlation. We also present many results from MUSIC in the uncorrelated and nearly correlated cases for comparison. In Appendix A, we take a brief look at ESPRIT for the perfectly uncorrelated case. In Appendix B, we present some results from a variation of MUSIC called "spatially averaged" MUSIC that *can* distinguish between correlated signals. Indeed, the results are comparable to Prony in the case of perfect correlation. Finally, in Appendix C, we glance at some of the recent literature on MUSIC.

2. THEORY

We will now describe the theory behind the first two algorithms examined in this report: the Prony algorithm [1] and the MUSIC algorithm [1, 2, 4]. In all cases we assume that the receiving antenna is in the far field of a limited number of discrete narrowband sources, so that at any given instant, the incoming signal is closely approximated by a set of discrete plane waves coming from different directions of arrival (DOA). (The definition of narrowband in this case is that the time of travel of a wavefront across the array is much less than the duration of the pulse.) We consider the one-dimensional case, i.e., the array is one-dimensional and the angle of arrival varies in the horizontal plane only, measured with respect to boresight. For simplicity, we ignore coupling among the elements, and the elements are assumed to all be the same.

Manuscript approved April 7, 1992.

*Prony's original algorithm was an exact fit to the noiseless case. With overdetermined equations, we can make a least-squares fit to the noisy case. This is equivalent to the popular Modified Covariance algorithm of spectral estimation [1, p. 314].

At any given instant, there is a set of complex baseband outputs across the elements, which will be referred to as a *snapshot*. A set of discrete snapshots is recorded over some relatively short interval of time during which the number of signals and their angles of arrival are assumed to remain essentially constant. Also, the statistics of the received signals must remain stationary during this interval. This includes the correlations between the (complex) amplitudes of the different plane waves (the different signals may have randomly varying amplitudes and/or initial phases from snapshot to snapshot), the correlations between the receiver noise at the elements (each element has its own receiver and the antenna is assumed to be internally noise limited), and the correlations between the (complex) amplitudes and the noise at the elements. (See Ref. 5 for further explanation.)

2.1 Prony Algorithm

The Prony algorithm models the signal across the elements as a sum of complex exponentials. This model is correct only when the array is linear, and therefore the Prony algorithm is *geometry-dependent* (unlike MUSIC and other eigenspace-based algorithms). In this respect, it is inferior to MUSIC; however, it has the advantage over MUSIC of being able to resolve perfectly correlated signals (which may occur, for example, in the case of multipath). Also, the Prony algorithm degenerates when noise is added, as this again implies a violation of the assumed signal model. This signal, for a given snapshot, is represented as

$$s_n(t_j) = \sum_{m=1}^{K} c_m(t_j)\, e^{in\,(kD \sin\theta_m)} \quad n = 0,1,2\ \cdots \tag{1}$$

where t_j is the time of the j^{th} snapshot, s_n is the value at the n^{th} element, K is the number of incoming signals (narrowband plane waves), c_m is the instantaneous amplitude of the m^{th} signal at the 0^{th} element (the element pattern is lumped into this constant), n is the element number, k is the wavenumber, D is the spacing between elements ($\lambda/2$ in all cases considered in this report), and θ_m is the angle of arrival of the m^{th} signal.

With the following compact notation

π

$$\begin{aligned} v_m &= kD \sin\theta_m \\ \underline{C}^T &= [c_1\ c_2\ \cdots\ c_K]^T \\ \underline{d}_n^T &= [e^{iv_1 n}\ e^{iv_2 n}\ \ldots\ e^{iv_K n}]^T \end{aligned} \tag{2}$$

we can rewrite the first equation as

$$s_n = \underline{C}^T\, \underline{d}_n = \sum_{m=1}^{K} c_m(t_j)\, e^{in\,(kD \sin\theta_m)} = \sum_{m=1}^{K} c_m(t_j)\, e^{inv_m}\,. \tag{3}$$

By introducing a shifting operator

$$\mathbf{D} = \begin{bmatrix} e^{iv_1} & 0 & \cdots & 0 \\ 0 & e^{iv_2} & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & \cdots & e^{iv_k} \end{bmatrix}, \tag{4}$$

we can write

$$\begin{aligned} s_n &= \underline{C}^T \underline{d}_n \\ s_{n-1} &= \underline{C}^T \underline{d}_{n-1} = \underline{C}^T \mathbf{D}^{-1} \underline{d}_n \\ &\cdots \\ s_{n-j} &= \underline{C}^T \mathbf{D}^{-j} \underline{d}_n \end{aligned} , \tag{5}$$

which can be written more concisely as

$$\underline{X}_n = \boldsymbol{\xi}\, \underline{d}_n \tag{6}$$

with

$$\underline{X}_n^T = [s_n\ s_{n-1}\ s_{n-2}\ \cdots\ s_{n-(L-1)}] \tag{7}$$

and

$$\boldsymbol{\xi} = \begin{bmatrix} c_1 & c_2 & \cdots & c_K \\ c_1 e^{-iv_1} & c_2 e^{-iv_2} & \cdots & c_K e^{-iv_K} \\ c_1 e^{-i2v_1} & c_2 e^{-i2v_2} & \cdots & c_K e^{-i2v_K} \\ \vdots & \vdots & \ddots & \vdots \\ c_1 e^{-i(L-1)v_1} & c_2 e^{-i(L-1)v_2} & \cdots & c_K e^{-i(L-1)v_K} \end{bmatrix} . \tag{8}$$

(L is some integer such that L-1 ≤ n ≤ N-1 and K<L. Its purpose will become clearer below.)

Now any rectangular matrix can be given a *singular value decomposition* (SVD) [1]. By applying this to the matrix $\boldsymbol{\xi}$, we can eventually arrive at an algorithm to determine the directions of arrival θ_m, as follows:

$$\boldsymbol{\xi} = \begin{bmatrix} \underline{C}^T \\ \underline{C}^T \mathbf{D}^{-1} \\ \cdot \\ \cdot \\ \underline{C}^T \mathbf{D}^{-L+1} \end{bmatrix} = \mathbf{U}_K\, \Sigma_K\, \mathbf{V}_K^H$$

$$\begin{aligned} \mathbf{U}_K^H\, \mathbf{U}_K &= \mathbf{1}_{KK} \\ \mathbf{U}_K\, \mathbf{U}_K^H &\neq \mathbf{1}_{KK} \\ \mathbf{V}_K\, \mathbf{V}_K^H &= \mathbf{1}_{KK} \end{aligned} \tag{9}$$

where $\mathbf{U}$ is an L by K matrix, $\boldsymbol{\Sigma}$ is a K by K diagonal matrix, and $\mathbf{V}$ is a K by K matrix. The diagonal elements of $\boldsymbol{\Sigma}$ are the singular values.

We can then use linear algebra to determine $\underline{d}_n$ in terms of $\underline{X}_n$

$$\underline{X}_n = (\mathbf{U}_K \, \mathbf{\Sigma}_K \, \mathbf{V}_K^H) \, \underline{d}_n$$

(substitution for ξ)

$$\mathbf{U}_K^H \, \underline{X}_n = (\mathbf{U}_K^H \, \mathbf{U}_K) \, \mathbf{\Sigma}_K \, \mathbf{V}_K^H \, \underline{d}_n = \mathbf{1}_{KK} \, \mathbf{\Sigma}_K \, \mathbf{V}_K^H \, \underline{d}_n \tag{10}$$

(premultiply by $\mathbf{U}_K^H$)

$$\mathbf{V}_K \, \mathbf{\Sigma}_K^{-1} \, \mathbf{U}_K^H \, \underline{X}_n = (\mathbf{V}_K \, \mathbf{V}_K^H) \, \underline{d}_n = \underline{d}_n$$

(premultiply by $\mathbf{V}_K \, \mathbf{\Sigma}_K^{-1}$)

which gives

$$\underline{d}_n = \xi^{-1} \, \underline{X}_n \tag{11}$$

where ξ^{-1} is the *Moore-Penrose pseudoinverse* defined by

$$\xi^{-1} = \mathbf{V}_K \, \mathbf{\Sigma}_K^{-1} \, \mathbf{U}_K^H . \tag{12}$$

Using this expression, we can then determine a *one-step predictor*, to express $\underline{X}_{n+1}$ in terms of $\underline{X}_n$

$$\underline{X}_{n+1} = \xi \, \mathbf{D} \, \xi^{-1} \, \underline{X}_n = \mathbf{A} \, \underline{X}_n$$

(substitution into $X_{n+1} = \xi \, \underline{d}_{n+1} = \xi \, \mathbf{D} \underline{d}_n$) (13)

where $\mathbf{A} = \xi \, \mathbf{D} \, \xi^{-1}$ is called the *system matrix.*

The system matrix satisfies the following *eigenvalue equation*

$$\begin{aligned} \mathbf{A} \, \xi_i &= \lambda_i \, \xi_i \\ \lambda_i &= e^{i v_i} \quad i = 1, 2, \ldots K \\ \lambda_i &= 0 \quad\quad i = K+1, \ldots L \end{aligned} \tag{14}$$

as can be determined by simple substitution for **A** and the fact that the pseudoinverse is a left (though not right) inverse to itself:

$$\mathbf{A} \, \xi = \xi \, \mathbf{D} \, (\xi^{-1} \, \xi) = \xi \, \mathbf{D} .$$

(remember that **D** is diagonal) (15)

(Notice that the directions of arrival are found from the eigenvalues; they are the arcsines of v_i /kD for i = 1,...K.)

We can represent the system matrix as follows

$$\underline{X}_{n+1} = \mathbf{A}\ \underline{X}_n \quad (L-1 < n < N-1)$$

$$\begin{bmatrix} s_{n+1} \\ s_n \\ s_{n-1} \\ . \\ . \\ . \end{bmatrix} = \begin{bmatrix} A_1 & A_2 & A_3 & \cdots & A_L \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ . & . & . & \cdots & . \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} = \begin{bmatrix} s_n \\ s_{n-1} \\ s_{n-2} \\ . \\ . \\ . \end{bmatrix}$$

$$s_{n+1} = \sum_{k=1}^{L} A_k\, s_{n-(k-1)} \Rightarrow s_{m+L} = \sum_{k=1}^{L} A_k\, s_{m+L-k} \qquad m = 0,1,2,\ldots N-L-1\,. \tag{16}$$

(The form of the matrix is clear from the following considerations: The first row of the matrix consists of the coefficients that give s_{n+1} as a linear combination of the previous L s-values, and the remaining rows of the matrix are simply identities among the s_n.)

The A_k can be solved in a least squares sense by the *normal equations* of linear algebra. Specifically, the last set of equations above has the form

$$\underline{x} = \mathbf{C}\,\underline{y} \qquad \text{(with } \underline{y} \text{ corresponding to the } A_k\text{)} \tag{17}$$

and an optimal (in the least-squares sense) $\underline{y}$ can be found by minimizing the functional $\underline{x} - \mathbf{C}\,\underline{y}$ with respect to $\underline{y}$

$$\min |\underline{x} - \mathbf{C}\,\underline{y}|^2 \Rightarrow \mathbf{C}^H \underline{x} = \mathbf{C}^H \mathbf{C}\,\underline{y}\,, \tag{18}$$

which can be re-expressed as

$$\sum_{m=0}^{N-L-1} s^*_{m+L-j}\, s_{m+L} = \sum_{k=1}^{L} \sum_{m=0}^{N-L-1} (s^*_{m+L-j}\, s_{m+L-k})\, A_k\,. \tag{19}$$

By defining the *correlation matrix*

$$\mathbf{R} = \sum_{n=L}^{N-1} \underline{X}^*_n\, \underline{X}^T_n$$
$$\underline{r}^T = [R_{00}\, R_{10} \cdots R_{N-L-1,\,0}] \tag{20}$$

and the vector

$$\underline{A} = [A_1\, A_2\, A_3 \cdots A_L] \tag{21}$$

we can rewrite the above equations concisely as

$$R_{j0} = \sum_{k=1}^{L} R_{jk}\, A_k \tag{22}$$

or

$$\underline{r} = \mathbf{R}\,\underline{A}\,. \tag{23}$$

We can then solve for the vector $\underline{A}$

$$\underline{A} = \mathbf{R}^{-1}\,\underline{r} \qquad \text{(provided, of course, that } \mathbf{R} \text{ is non-singular)} \tag{24}$$

which readily gives the system matrix **A**. By solving the eigenvalue equation of the system matrix, we can finally determine the directions of arrival from the eigenvalues, as described above.

An alternative solution that lessens the computational load by projecting the problem onto a smaller subspace is as follows. The correlation matrix has an SVD decomposition

$$\mathbf{R} = \mathbf{V}_K\,\mathbf{\Sigma}_K\,\mathbf{V}_K^H \tag{25}$$

where the right-hand matrix is the Hermitian of the left-hand matrix because **R** itself is Hermitian. We then project the data vectors onto the K-dimensional subspace spanned by the singular vectors that comprise the columns of $\mathbf{V}_K$

$$\underline{\tilde{X}}_n = \mathbf{V}_K^T\,\underline{X}_n \tag{26}$$

which gives a reduced correlation matrix

$$\tilde{\mathbf{R}} = \sum_{n=L}^{N-1} \underline{\tilde{X}}_n^*\,\underline{\tilde{X}}_n^T = \sum_{n=L}^{N-1} \mathbf{V}_K^H\,\underline{X}_n^*\,\underline{X}_n^T\,\mathbf{V}_K = \mathbf{V}_K^H\,\mathbf{R}\,\mathbf{V}_K = \mathbf{\Sigma}_K$$
$$\text{(the last step results from various forms of the relation } \mathbf{V}_K^H\,\mathbf{V}_K = \mathbf{1}_{KK}) \tag{27}$$

equal to the matrix of singular values

$$\tilde{\mathbf{R}} = \mathbf{\Sigma}_K\,. \tag{28}$$

We can then determine a reduced one-step prediction (or system) matrix:

$$\underline{X}_{n+1} = \mathbf{A}\,\underline{X}_n \qquad \text{(given previously)} \tag{29}$$

$$\mathbf{V}_K^*\,\underline{\tilde{X}}_{n+1} = \mathbf{A}\,\mathbf{V}_K^*\,\underline{\tilde{X}}_n \qquad \text{(from the definition of the reduced data vector)} \tag{30}$$

$$\mathbf{V}_K^T\,\mathbf{V}_K^*\,\underline{\tilde{X}}_{n+1} = \mathbf{V}_K^T\,\mathbf{A}\,\mathbf{V}_K^*\,\underline{\tilde{X}}_n \qquad \text{(premultiplying by } \mathbf{V}_K^T) \tag{31}$$

$$\underline{\tilde{X}}_{n+1} = \tilde{\mathbf{A}}\,\underline{\tilde{X}}_n$$
$$\tilde{\mathbf{A}} = \mathbf{V}_K^T\,\mathbf{A}\,\mathbf{V}_K^*\,. \qquad \text{(and using } \mathbf{V}_K^T\mathbf{V}_K{}^* = \mathbf{1}_{KK}) \tag{32}$$

Now defining the matrix $\mathbf{R}_1$

$$\mathbf{R}_1 = \sum_{n=L-1}^{N-2} \tilde{\underline{X}}_{n+1}^{*} \tilde{\underline{X}}_n^{T} = \sum_{n=L-1}^{N-2} \tilde{\mathbf{A}}^{*} \tilde{\underline{X}}_n^{*} \tilde{\underline{X}}_n^{T} = \tilde{\mathbf{A}}^{*} \tilde{\mathbf{R}} = \tilde{\mathbf{A}}^{*} \mathbf{\Sigma}_K$$

(every step a substitution of a previous identity) (33)

we can express the reduced system matrix as

$$\tilde{\mathbf{A}} = \mathbf{R}_1^{*} \mathbf{\Sigma}_K^{-1} .$$

(taking the conjugate of both sides and using the diagonality of $\mathbf{\Sigma}_K$) (34)

It can be shown that the reduced system matrix above has the same nonzero eigenvalues as the full system matrix. However, its dimension is only K by K, so that to find the eigenvalues, we need solve only a K-dimensional characteristic polynomial instead of a possibly much larger one. (The eigenvalues, we remember, give the directions of arrival.)

$$\tilde{\mathbf{A}} \tilde{\xi}_i = \lambda_i \tilde{\xi}_i$$

$$\lambda_i e^{i\nu_i} \quad (i = 1,..., K) \qquad \tilde{\xi}_i = \mathbf{V}_K^{T} \xi_i . \tag{35}$$

In the case of *noise*, the Prony method degenerates, as the assumed sinusoidal model only approximates the data. We now need some kind of averaging, and there are different ways of doing this. (We assume that the noise is zero-mean and stationary.)

The *Prony space* method calculates the directions of arrival (from the eigenvalues) separately for each snapshot and then averages them over all the snapshots. Because of the assumed stationarity and zero mean of the noise, the averaged estimates of the directions of arrival should asymptotically approach the true values. In averaging the DOAs, we must assume that the rank K (the number of singular values) remains constant from snapshot to snapshot, or else we must make some provision to account for changing rank. Also, the variance of the eigenvalues cannot be so great that we cannot identify related eigenvalues from snapshot to snapshot. However, if the noise is this great, the method probably would not work anyway.

The *Prony space-time* method, on the other hand, first averages the relevant matrices over all snapshots and then determines the eigenvalues from the averaged matrices. There are different ways of doing this. At first, I averaged $\mathbf{R}$ and $\mathbf{R}_1$, performed an SVD on the averaged $\mathbf{R}$, calculated the inverse of $\mathbf{\Sigma}$ from these singular values, and then used this inverse along with the average of $\mathbf{R}_1$ in the equation

$$\tilde{\mathbf{A}} = \mathbf{R}_1^{*} \mathbf{\Sigma}_K^{-1} \tag{36}$$

to find the reduced system matrix. Then I calculated the eigenvalues from this reduced system matrix.

This did not work very well. (It gave poorer results than the Prony space algorithm, and averaging did not seem to help much.) I then tried an alternative technique, which seems to work somewhat better than the Prony space algorithm, and the results of which are presented in this report. This consisted simply of averaging $\mathbf{R}_1$ and the inverse of $\mathbf{\Sigma}$ calculated from each snapshot, and then using these two averaged matrices in the equation above. (I did not use the average of the correlation matrix $\mathbf{R}$.)

2.2 MUSIC Algorithm

The MUSIC algorithm has two advantages over the Prony algorithm. It is geometry-independent (i.e., it does not require the array to have a specific geometry) and it incorporates noise into its mathematical structure to reduce its effects. However, unlike Prony, it breaks down in the case of perfectly or nearly perfectly correlated signals, and this case is often of interest (for example, in distinguishing multipath from true targets). (The precise definition of "correlated signals" will become clear in the discussion.)

MUSIC starts by explicitly incorporating noise into its signal model

$$z_n(t_j) = \sum_{m=1}^{K} W_n(\theta_m)\, c_m(t_j) + \nu_n(t_j) \quad n = 0, 1, \ldots, N-1 . \tag{37}$$

The various quantities are as follows: z_n is the signal at element n at time t_j. c_m is the complex amplitude of plane wave m at reference element 0 at time t_j. $W_n(\theta_m)$ is the steering vector (at element n in direction θ_m) that accounts for both the delays of the various plane waves between the reference element and element n and also any weights that may be applied to the elements. (It is the explicit incorporation of this steering vector that allows independence from geometry.) Finally, ν_n is the noise at element n at time t_j. It is assumed that the noise is uncorrelated from element to element and is also uncorrelated with the signal.

The signal model can be written more concisely as

$$\underline{z} = \mathbf{W}\,\underline{c} + \underline{\nu} \tag{38}$$

where $\underline{z}$, $\underline{c}$, and $\underline{\nu}$ are vectors and $\mathbf{W}$ is a matrix (as are all boldface quantities in what follows).

We can define the *correlation matrix* of the received signal

$$\mathbf{R} = \langle \underline{z}\,\underline{z}^H \rangle = \mathbf{W} \langle \underline{c}\,\underline{c}^H \rangle \mathbf{W}_H = \mathbf{W}\,\mathbf{C}\,\mathbf{W}_H + \sigma^2\,\mathbf{I} \tag{39}$$

where brackets denote the statistical ensemble average. This correlation matrix has a special structure, the exploitation of which is an integral part of the MUSIC algorithm. First, because the steering vectors are deterministic quantities, the brackets pass through them to the (stochastic) signal amplitudes on the inside, thus defining the *power matrix* $\mathbf{C}$. If the signal amplitudes are uncorrelated among themselves, then $\mathbf{C}$ has full rank K, and as we will see, this is a necessary condition for MUSIC to work. Since the noise is assumed uncorrelated from element to element, the brackets give an N by N (K<N) diagonal matrix for the correlation of the noise with itself, and since the noise is assumed stationary over the array (as well as over the interval of observation) this diagonal matrix is simply the identity matrix multiplied by a scalar, the noise power σ^2. Since the noise is uncorrelated with the signal, there are no cross terms between noise and signal.

The correlation matrix $\mathbf{R}$ satisfies an eigenvalue equation

$$\mathbf{R}\,\underline{\xi}_j = \lambda_j\,\underline{\xi}_j \quad j = 0, \ldots, N-1 . \tag{40}$$

These eigenvalues have an important property, namely, that N-K of them are degenerate and equal to the noise power

$$\lambda_j = \sigma^2 \quad j = K+1, \ldots, N-1 . \tag{41}$$

Furthermore, this degenerate eigenvalue is the *minimum eigenvalue*; all others are greater. The argument is based on the rank structure of the correlation matrix **R**. The signal part of this matrix ($\mathbf{WCW}_H$) has rank K (due to a theorem of linear algebra that says that the rank of a product of three matrices is controlled by the one in the middle), and so this term can only contribute towards K of the eigenvalues. The noise part of the correlation matrix, however, has full rank N, and so contributes to all eigenvalues. (Being σ^2 times the unit matrix, this contribution is just σ^2.) Furthermore, the signal part of the correlation matrix contributes only positive amounts to the eigenvalues (in addition to the σ^2 from the noise), because it is Hermitian and positive definite. (A Hermitian matrix has only real eigenvalues, and a positive definite matrix has positive eigenvalues.) So the noise eigenvalues are the minimum eigenvalues.

With this assumption on the eigenvalues, we can write

$$\begin{aligned} \mathbf{R}\,\underline{\xi}_j = \mathbf{W}\,\mathbf{C}\,\mathbf{W}^H\,\underline{\xi}_j + \sigma^2\,\underline{\xi}_j = \sigma^2\,\underline{\xi}_j \qquad & (j = K+1, \ldots, N-1) \\ \Rightarrow\ \mathbf{W}\,\mathbf{C}\,\mathbf{W}^H\,\underline{\xi}_j = 0 \qquad & (j = K+1, \ldots, N-1) \\ \Rightarrow\ \mathbf{W}^H\,\underline{\xi}_j = 0 \qquad & (j = K+1, \ldots, N-1) \\ \Rightarrow\ \mathbf{W}^H\,\underline{E}_j = 0 \qquad & \underline{E}_j = \underline{\xi}_j \quad (j = K+1, \ldots, N-1)\,. \end{aligned} \tag{42}$$

The next to last step is true only if the matrix **C** has full rank (so that it is nonsingular and can be inverted), and since this is the key to the MUSIC method, we see why it is important for the signals to be uncorrelated.

We can therefore decompose the eigenspace into a K-dimensional *signal subspace* spanned by the eigenvectors associated with the K eigenvalues that are greater than σ^2, and an N-K dimensional *noise subspace* of eigenvectors $\underline{E}_j$ that are orthogonal to the steering vector (as indicated above).

This orthogonality occurs only when the steering vector is in fact steering in the direction of the signal, as assumed in the construction above. Therefore, we can calculate or estimate the steering vector for a whole grid of directions (this is sometimes called the *array manifold*), solve the eigenvalue problem of the correlation matrix, and search over the array manifold for the directions that are orthogonal to the noise eigenvectors. To do this, we must first estimate the correlation matrix from the data samples. Averaging over snapshots will asymptotically lead to the true correlation matrix (by ergodicity), but of course we cannot average for an interval longer than that for which the target and noise characteristics remain stationary. The averaged data correlation matrix has the form

$$\mathbf{R} \approx \frac{1}{M} \sum_{j=1}^{M} \underline{z}(t_j)\ \underline{z}^H(t_j)\,. \tag{43}$$

A MUSIC pseudospectrum can be generated by the following expression

$$\begin{aligned} S(\theta) &= \frac{1}{\underline{W}^H(\theta)\,\mathbf{E}\,\mathbf{E}^H\,\underline{W}(\theta)} \\ \mathbf{E} &= [\underline{E}_{K+1} \cdots \underline{E}_{N-1}]\,. \end{aligned} \tag{44}$$

Note that sharp peaks occur when the denominator is zero, i.e., when the orthogonality condition is reached. The denominator is actually (the square of) the average over all noise

eigenvectors of the equation $\mathbf{W}^H \underline{E}_j = 0$. This reduces the effect of *spurious* zeros that do not represent a true signal but that may arise when the dimension of the noise space is greater than the number of signals. The true zeros persist over the averaging because they satisfy the orthogonality condition, but the accidental zeros do not.

3. MUSIC RESULTS

The MUSIC results include all figures beginning with the letters "A" and "B." They are organized as follows.

All cases have 11 elements with $\lambda/2$ spacing. All cases have two equal amplitude plane waves coming in: one at $\theta_1 = 45°$ and the other at $\theta_2 = 45°$ to 50° at 1/2° spacing. (All figures show the true angles of arrival as straight lines on the lower subplots.) Figures beginning with the letter "A" all have perfect decorrelation between the signals ($\alpha = 1.0$), and figures beginning with the letter "B" consider cases at or near perfect correlation ($\alpha = 0.5$ for perfect correlation).

Note: The signals are generated as follows (for $\lambda/2$ spacing):

$$S_1(\text{JSNAP, IELEM}) = e^{j(\text{JSNAP-1})\frac{\pi}{16}} e^{j(\text{IELEM-1})\,\pi \sin\theta_1} \tag{45}$$

$$S_2(\text{JSNAP, IELEM}) = e^{j\,\alpha(\text{JSNAP-1})\frac{\pi}{8}} e^{j(\text{IELEM-1})\,\pi \sin\theta_2} \tag{46}$$

$$S_{\text{total}} = S_1 + S_2 + \nu\ (\text{JSNAP,IELEM}) \tag{47}$$

where S_1 is the first plane wave for snapshot JSNAP at element IELEM, S_2 is the second plane wave, and ν is the white Gaussian uncorrelated noise* . In this case, for $\alpha = 1.0$, the signals ("in space," see Ref. 5) are uncorrelated due to Fourier orthogonality. (Statistical orthogonality, i.e., decorrelation, is a much more general stochastic concept that includes Fourier orthogonality as a particular case, at least for the ergodic case where ensemble averages are equal to integrals over time.)

3.1 Explanation of MUSIC Figures

The "A" group is divided into four series, which we will refer to as A1 (Figs. A1.1 to A1.11), A2 (Figs. A2.1 to A2.8), A3 (Figs. A3.1 to A3.8), and A4 (Figs. A4.1 to A4.8).

- Series A1 considers the case of 64 snapshots for various SNRs. The SNRs considered are: 5, 7, 8, 9, 10, 15, 20, 30, 40, 50, and 99 dB.
- Series A2 considers the case of 10 dB SNR for various numbers of snapshots. The numbers of snapshots considered are: 1, 2, 5, 10, 20, 30, 64, and 100 snapshots.
- Series A3 considers the case of 20 dB SNR for various numbers of snapshots. The numbers of snapshots considered are: 1, 2, 5, 10, 20, 30, 64, and 100 snapshots.

---

* The noise is generated according to a standard algorithm for converting uniformly distributed random variables (as provided by the random number generator of a computer) to white Gaussian random variables. A and B are random variables uniformly distributed in [0,1]. The variance s^2 is derived (for unity amplitude signals) as

$$\sigma^2 = \frac{1}{2 \cdot 10^{\frac{SNR_{dB}}{10}}}.$$

With $R = \sqrt{2\sigma^2 \ln\left(\frac{1}{1-A}\right)}$ and $\Theta = 2\pi B$ a sample of complex white Gaussian noise ν is given by $Re^{i\Theta}$.

- Series A4 considers the case of 30 dB SNR for various numbers of snapshots. The numbers of snapshots considered are: 1, 2, 5, 10, 20, 30, 64, and 100 snapshots.

The "B" group is divided into two series, which we will refer to as B1 (Figs. B1.1 to B1.7) and B2 (Figs. B2.1 to B2.7).

- Series B1 considers the case of 64 snapshots and 10 dB SNR for various α's. The α's considered are: 0.50 (ρ = 1.00), 0.55 (ρ = 0.94), 0.56 (ρ = 0.91), 0.57 (ρ = 0.88), 0.58 (ρ = 0.84), 0.59 (ρ = 0.80), and 0.60 (ρ = 0.76).
- Series B2 considers the case of 64 snapshots and 20 dB SNR for various α's. The α's considered are: 0.50 (ρ = 1.00), 0.51 (ρ = 1.00), 0.52 (ρ = 0.99), 0.53 (ρ = 0.98), 0.54 (ρ = 0.96), 0.55 (ρ = 0.94), and 0.56 (ρ = 0.91).

All MUSIC figures have the same format: an upper plot with a MUSIC pseudospectrum, and a lower plot representing the estimates of the angles of arrival. Note that the upper plot also contains the tops of the broad peaks resulting from an unwindowed FFT of the datapoints in the first of the snapshots from the set of snapshots that generated the corresponding MUSIC spectrum. This is to indicate the dramatic improvement in resolution permitted by the high-resolution algorithms. The estimates of the angles of arrival in the lower plot were found simply by searching for local maxima in the MUSIC pseudospectrum. The resolution of the MUSIC spectrum is 256 points over the angular range in the axis of the spectral plot. The true angles of arrival are indicated by the straight lines in the lower plot.

3.1.1 Series A1 (α = 1.00; 64 snaps; various SNRs)

The general trend is that the signals are resolved at an increasingly lower angular separation with increasingly high SNR, until at 50 dB SNR and above, the estimates are essentially perfect. There is bias in the estimates near the angular threshold of resolution, as one might expect (one apparent peak gradually splits into two), but the estimates rapidly converge to the true values above this threshold. Interestingly, the lower angle (i.e., 45°) quickly converges to its exact value, while the upper angle keeps a slight bias over the angular range indicated until the SNR reaches 40 dB. This asymmetry has nothing to do with the fact that the lower angle is held constant, because the estimates are made separately for the different angular separations. The resolution thresholds for the various SNRs are:

- 8 dB SNR: β^* = 0.27 (4.6°) (of angular separation)
- 9 dB SNR: β = 0.24 (4.0°)
- 10 dB SNR: β = 0.24 (4.0°)
- 15 dB SNR: β = 0.18 (3.0°)
- 20 dB SNR: β = 0.12 (2.0°)
- 30 dB SNR: β = 0.06 (1.0°)
- 40 dB SNR: β = 0.03 (0.5°)
- 50 dB SNR: β = 0.00 (0.0°)
- 99 dB SNR: β = 0.00 (0.0°).

3.1.2 Series A2 (α = 1.00; 10 dB SNR; various numbers of snapshots)

Resolution is not attained (over the angular range of these plots) until 20 snapshots. With more snapshots, the resolution threshold tends to decrease, although at 64 snapshots it increased over the previous case, showing that more snapshots does not necessarily result in an improvement. At 30 snapshots, the lower angle rapidly converges to its true

*β is a measure of angular separation of targets that takes into account the reduced effective aperture of the array as we scan away from boresight: $\beta = 5\,[\sin\theta_2 - \sin\theta_1]$.

value, but the upper angle maintains a slight bias, and this pattern is repeated even at 100 snapshots. This parallels the trend observed in Series A1, in which the upper angle keeps a bias (until high SNR) while the lower angle does not. The resolution thresholds for the various numbers of snapshots are:

- 20 snapshots: $\beta = 0.24$ (4.0°)
- 30 snapshots: $\beta = 0.21$ (3.5°)
- 64 snapshots: $\beta = 0.24$ (4.0°)
- 100 snapshots: $\beta = 0.18$ (3.0°).

3.1.3 Series A3 ($\alpha = 1.00$; 20 dB SNR; various numbers of snapshots)

Resolution is not attained (over the angular range of these plots) until 10 snapshots. With more snapshots, the resolution threshold tends to decrease, at first quickly, then much more slowly. Again, the upper angle tends to have slightly more bias over all cases than the lower angle, although we can say that the estimates are very good above the threshold for 20 or more snapshots. The resolution thresholds for the various numbers of snapshots are:

- 10 snapshots: $\beta = 0.27$ (4.5°)
- 20 snapshots: $\beta = 0.15$ (2.5°)
- 30 snapshots: $\beta = 0.12$ (2.0°)
- 64 snapshots: $\beta = 0.12$ (2.0°)
- 100 snapshots: $\beta = 0.09$ (1.5°).

3.1.4 Series A4 ($\alpha = 1.00$; 30 dB SNR; various numbers of snapshots)

Resolution is attained (over the angular range of these plots) at just 5 snapshots. With more snapshots, the resolution threshold tends to decrease, at first quickly, then more slowly. The upper angle no longer tends to have slightly more bias than the lower angle. The estimates are good for 10 snapshots and very good for 20 or more snapshots, with rapid convergence above threshold to the true values. The resolution thresholds for the various numbers of snapshots are:

- 5 snapshots: $\beta = 0.24$ (4.0°)
- 10 snapshots: $\beta = 0.15$ (2.5°)
- 20 snapshots: $\beta = 0.06$ (1.0°)
- 30 snapshots: $\beta = 0.06$ (1.0°)
- 64 snapshots: $\beta = 0.06$ (1.0°)
- 100 snapshots: $\beta = 0.03$ (0.5°).

3.1.5 Series B1 (64 snapshots; 10 dB SNR; various α's)

Resolution is attained (over the angular range of these plots) at $\alpha = 0.57$ ($\rho = 0.88$). The resolution threshold decreases somewhat with higher α (although we do not consider α's higher than 0.60). The resolution thresholds for the various α's are:

- $\alpha = 0.57$ ($\rho = 0.88$): $\beta = 0.27$ (4.5°)
- $\alpha = 0.58$ ($\rho = 0.84$): $\beta = 0.24$ (4.0°)
- $\alpha = 0.59$ ($\rho = 0.80$): $\beta = 0.24$ (4.0°)
- $\alpha = 0.60$ ($\rho = 0.76$): $\beta = 0.24$ (4.0°).

3.1.6 Series B2 (64 snapshots; 20 dB SNR; various α's)

Resolution is attained (over the angular range of these plots) at $\alpha = 0.53$ ($\rho = 0.98$). The resolution threshold decreases somewhat with higher α (although we do not consider α's higher than 0.56). The resolution thresholds for the various α's are:

- $\alpha = 0.53$ ($\rho = 0.98$): $\beta = 0.24$ (4.0°)
- $\alpha = 0.54$ ($\rho = 0.96$): $\beta = 0.21$ (3.5°)
- $\alpha = 0.55$ ($\rho = 0.94$): $\beta = 0.21$ (3.5°)
- $\alpha = 0.56$ ($\rho = 0.91$): $\beta = 0.18$ (3.0°).

3.2 MUSIC: Conclusions

3.2.1 Perfectly uncorrelated case

The resolution properties of the MUSIC algorithm are excellent in the perfectly uncorrelated case ($\alpha = 1.0$). The resolution threshold decreases with both increased SNR and increased number of snapshots, with SNR being the more important factor. For 64 snapshots, a resolution of $\beta = 0.12$ (2.0°) is achieved at 20 dB SNR and of $\beta = 0.06$ (1.0°) at 30 dB SNR. Above 50 dB SNR, the estimates are essentially perfect. For 10 dB SNR, resolution is attained at $\beta = 0.24$ (4.0°) with 20 snapshots. For 20 dB SNR, resolution is attained with 10 snapshots, but is much better with 20 or more snapshots (1.5° to 2.5°). For 30 dB SNR, resolution is attained with 5 snapshots, but is much better with 10 or more snapshots (2.5° for 10 snapshots; 0.5° to 1.0° for 20 or more snapshots). Above the resolution threshold, the estimates rapidly converge to the true values, especially with 20 or more snapshots at SNRs of 20 dB or higher. At lower SNR and fewer snapshots, there is a tendency towards more bias in the upper angle than in the lower one (for the angular cases considered here).

3.2.2 Nearly correlated cases

The MUSIC algorithm can achieve resolution at surprisingly high correlation, considering the theoretical (and actual) breakdown at perfect correlation. For 10 dB SNR and 64 snapshots, resolution is attained at $\alpha = 0.57$ ($\rho = 0.88$), although the angular separation (4.5°) is probably too large to be considered "high resolution." (This separation does not decrease as α approaches 0.60 ($\rho = 0.76$), the maximum considered here.) For 20 dB SNR and 64 snapshots, resolution is attained at $\alpha = 0.53$ ($\rho = 0.98$), with an angular separation of 4.0°. This separation decreases to 3.0° as α approaches 0.56 ($\rho = 0.91$), the maximum considered here.

4. PRONY RESULTS

The Prony results include all figures beginning with the letter "C." They are organized as follows.

All cases have 11 elements with $\lambda/2$ spacing. All cases have two equal amplitude plane waves coming in: one at $\theta_1 = 45°$ and the other at $\theta_2 = 45°$ to 60° at 1/2° spacing. (All figures show the true angles of arrival as straight lines.) Most importantly, all cases consider the perfect correlation case ($\alpha = 0.5$), since, as we have seen, such eigenvector techniques as MUSIC work very well when just a little decorrelation is present. It is in the case of perfect correlation (which may occur, for example, with multipath) that makes the Prony technique particularly interesting. We consider both the "Prony space" and the "Prony space-time" variations of the algorithm, as described previously in this report.

4.1 Explanation of Prony Figures

There are four series, which we will refer to as: C1 (Figs. C1.1 to C1.9), C2 (Figs. C2.1 to C2.9), C3 (Figs. C3.1 to C3.9), and C4 (Figs. C4.1 to C4.9). Within a series, the parameter that varies is the number of snapshots. We will consider the following cases: 1, 2, 3, 5, 10, 20, 30, 64, and 100 snapshots. From series to series, the parameter that varies is the SNR (defined previously in this report). Series C1 considers 10 dB SNR; C2 considers 20 dB SNR; C3 considers 30 dB SNR; and C4 considers 40 dB SNR. In this way, we have a Cartesian product of SNR vs number of snapshots, with all other parameters maintained constant. Our objective is to resolve the signals with the fewest snapshots for a given SNR.

4.1.1 Series C1 (10 dB SNR)

In all cases, the Prony space algorithm shows a threshold at about 14° of angular separation between the plane waves ($\theta_1 = 45°$ and $\theta_2 = 59°$). This corresponds to $\beta = 0.75$. The threshold suddenly becomes sharp at 3 snapshots (Fig. C1.3), although we can say that the signals are resolved (with bias) at 1 and 2 snapshots. With 1 snapshot (Fig. C1.1) the bias at $\beta = 0.75$ is a few degrees, and with 2 snapshots (Fig. C1.2) no bias exists for Prony space at the upper angle ($\theta_2 = 59°$) and a couple of degrees of bias exist at the lower angle ($\theta_1 = 45°$). At 3 snapshots (Fig. C1.3), very little bias (at $\beta = 0.75$) is present on the lower angle, but there is a bias of about 5 degrees at the upper angle, which decreases appreciably with an increase in the number of snapshots.

The Prony space-time algorithm works slightly better, with about the same threshold of resolution ($\beta = 0.75$), but without the bias on the upper angle at the threshold itself. In fact, one could say that the Prony space-time algorithm generally resolves the signals at a much lower threshold (β @ 0.4 or 0.5), but with erratic and often large biases below the threshold of the Prony space algorithm.

4.1.2 Series C2 (20 dB SNR)

In all cases, the Prony space algorithm shows a threshold at about 8° of angular separation between the plane waves ($\theta_1 = 45°$ and $\theta_2 = 53°$). This corresponds to $\beta = 0.46$. The threshold suddenly becomes sharp at 3 snapshots (Fig. C2.3), although we can say that the signals are resolved (with bias) at 1 and 2 snapshots. With 1 snapshot (Fig. C2.1) the bias at $\beta = 0.46$ is a few degrees, and with 2 snapshots (Fig. C2.2) there is no bias for Prony space at the upper angle ($\theta_2 = 53°$) and a couple of degrees of bias at the lower angle ($\theta_1 = 45°$). At 3 snapshots (Fig. C2.3), there is no bias (at $\beta = 0.46$) on the lower angle, but a bias of about 7 degrees at the upper angle, which decreases appreciably with an increase in the number of snapshots. Above the threshold, the Prony space estimates rapidly converge to the true values.

The Prony space-time algorithm works slightly better, with about the same threshold of resolution ($\beta = 0.46$), but without such a large bias on the upper angle at the threshold itself. In fact, one could say that the Prony space-time algorithm generally resolves the signals at a much lower threshold (β @ 0.35 or even 0.25 in some cases), but with erratic and often large biases below the threshold of the Prony space algorithm. (The pattern of these erratic biases bears some similarity to those of the previous series.)

4.1.3 Series C3 (30 dB SNR)

In all cases, the Prony space algorithm shows a threshold at about 4.5° of angular separation between the plane waves ($\theta_1 = 45°$ and $\theta_2 = 49.5°$). This corresponds to $\beta = 0.27$. The threshold suddenly becomes sharp at 3 snapshots (Fig. C3.3), although we can

say that the signals are resolved (with bias) at 1 and 2 snapshots. With 1 snapshot (Fig. C3.1), the bias at $\beta = 0.27$ is a few degrees or less, and with 2 snapshots (Fig. C3.2), there is negligible bias for Prony space at the upper angle ($\theta_2 = 49.5°$) and only a degree of bias at the lower angle ($\theta_1 = 45°$). At 3 snapshots (Fig. C3.3), there is no bias (at $\beta = 0.27$) on the lower angle, but a bias of about 5 degrees at the upper angle, which decreases appreciably with an increase in the number of snapshots. Above the threshold, the Prony space estimates rapidly converge to the true values.

The Prony space-time algorithm works slightly better, with about the same threshold of resolution ($\beta = 0.27$), but without such a large bias on the upper angle at the threshold itself. In fact, one could say that the Prony space-time algorithm generally resolves the signals at a much lower threshold (β @ 0.20 or even 0.15 in some cases), but with erratic and often large biases below the threshold of the Prony space algorithm. (The pattern of these erratic biases bears some similarity to those of the two previous series.)

4.1.4 Series C4 (40 dB SNR)

In all cases, the Prony space algorithm shows a threshold at about 2.5° of angular separation between the plane waves ($\theta_1 = 45°$ and $\theta_2 = 47.5°$). This corresponds to $\beta = 0.15$. The threshold suddenly becomes sharp at 3 snapshots (Fig. C4.3), although we can say that the signals are resolved (with bias) at 1 and 2 snapshots. With 1 snapshot (Fig. C4.1), the bias at $\beta = 0.15$ is a few degrees or less, and with 2 snapshots (Fig. C3.2), there is negligible bias for Prony space at the upper angle ($\theta_2 = 47.5°$) and only a degree of bias at the lower angle ($\theta_1 = 45°$). At 3 snapshots (Fig. C4.3), there is no bias (at $\beta = 0.15$) on the lower angle, but a bias of about 5 degrees at the upper angle, which decreases appreciably with an increase in the number of snapshots. Above the threshold, the Prony space estimates rapidly converge to the true values.

The Prony space-time algorithm works slightly better, with about the same threshold of resolution ($\beta = 0.15$), but without such a large bias on the upper angle at the threshold itself. In fact, one could say that the Prony space-time algorithm generally resolves the signals at a much lower threshold (β @ 0.10 or even 0.09), but with erratic and often large biases below the threshold of the Prony space algorithm.

4.2 Prony: Conclusions

The different series of cases (different SNRs) just considered obviously exhibit a similar pattern. The Prony space algorithm show a well-defined threshold for good (i.e., unbiased) resolution that decreases with SNR: $\beta = 0.75$ (14° of angular separation) at 10 dB SNR; $\beta = 0.46$ (8° of angular separation) at 20 dB SNR; $\beta = 0.27$ (4.5° of angular separation) at 30 dB SNR; and $\beta = 0.15$ (2.5° of angular separation) at 40 dB SNR. This threshold always becomes sharp at just 3 snapshots, although in all cases we can say that we have resolved the signals (with bias) at just 1 snapshot. At 3 snapshots, there is always a bias of about 5° (at threshold) at the upper angle, which decreases appreciably with an increase in the number of snapshots, and there is little bias (at threshold) at the lower angle. For 20 dB or more of SNR, the estimates converge rapidly above the threshold to the true values, and even at 10 dB SNR, the biases are not large above the threshold. The Prony space-time algorithm works at least as well as the Prony space algorithm, and if one is willing to tolerate erratic and large biases near threshold, one can say that the Prony space-time algorithm shows an improvement in the β-threshold of 0.1 or even 0.2 over the Prony space algorithm (with the larger improvement at lower SNR).

Actually, it appears that with 3 snapshots, resolution (with bias) occurs in Prony space-time down to β=0 at all SNR's (10, 20, 30, and 40 dB). Although the bias is not small, this result may still be significant in that at least it tells us that another target is there, in

approximately a certain direction. One must still ask, though, whether this 'resolution' is spurious. Why, for instance, does it disappear with more snapshot averaging? Even though this phenomenon is curious, it seems difficult to believe that it is completely accidental.

5. CONCLUSIONS

Detailed conclusions for the MUSIC and Prony algorithms were provided in Sections 3 and 4, respectively.

MUSIC gives excellent resolution in all but the most correlated cases. Following the advice of the reviewer, we present in Appendix B some results from a variation of MUSIC called spatially averaged MUSIC that performs as well as Prony in the case of perfectly correlated signals. The resolution threshold decreases with both increased SNR and increased number of snapshots, with SNR being the more important factor. In the perfectly uncorrelated case, and for 64 snapshots, a resolution of $\beta = 0.12$ (2.0°) is achieved at 20 dB SNR and of $\beta = 0.06$ (1.0°) at 30 dB SNR. Above 50 dB SNR, the estimates are essentially perfect. For the nearly correlated case of $\alpha = 0.53$ (perfect correlation: $\alpha = 0.50$), an angular separation of 4.0° is attained for 20 dB SNR and 64 snapshots.

For the Prony space algorithm in the perfectly correlated case, there is a well-defined threshold for good (i.e., unbiased) resolution that decreases with SNR: 14° of angular separation at 10 dB SNR; 8° of angular separation at 20 dB SNR; 4.5° of angular separation at 30 dB SNR; and 2.5° of angular separation at 40 dB SNR. This threshold always becomes sharp at just 3 snapshots, although in all cases we can say that we have resolved the signals (with bias) at just 1 snapshot. Prony space-time seems to work even better. In some cases, the "resolution" is so good that one wonders if it is real.

In summary, the well-known excellent performance of non-spatially-averaged MUSIC in all but the most correlated cases has been verified, and it has been demonstrated that the Prony technique is promising in the perfectly correlated case. Also, the Prony technique seems to require few snapshots to resolve close targets.

ACKNOWLEDGMENTS

The derivations presented here of both the Prony and MUSIC algorithms are based on material learned in Professor Wasylkiwskyj's high-resolution array course, EE335, given at the George Washington University in the Fall of 1990. I also wish to express my appreciation to Dr. W.F. Gabriel for reviewing my report and making helpful suggestions.

REFERENCES

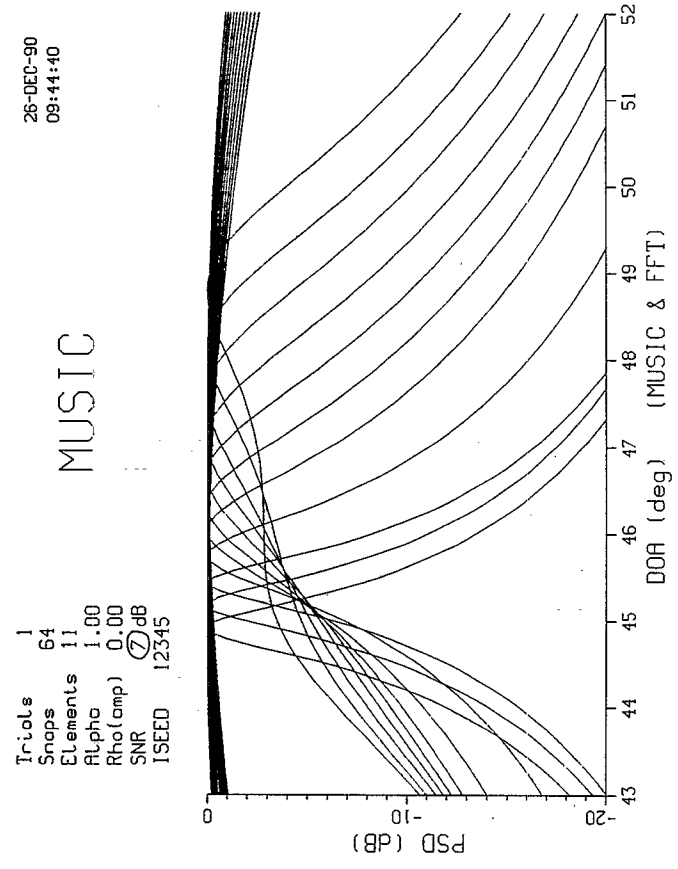
1. S. L. Marple, *Digital Spectral Analysis (*Prentice-Hall, Englewood Cliffs, N.J., 1987).

2. R. O. Schmidt, "Multiple Emitter Location and Signal Parameter Estimation," *IEEE Trans. Ant. Prop.*, **AP-34**, 276 (1986).

3. R. Roy and T. Kailath, "ESPRIT - Estimation of Signal Parameters Via Rotational Invariance Techniques," *IEEE Trans. Acous., Speech, Sig. Proc.*, **37**(7), 984 (1989).

4. S. J. Orphanidis, *Optimum Signal Processing* (Macmillan, New York, N.Y., 1988).

5. S. Haykin, ed., *Array Signal Processing* (Prentice-Hall, Englewood Cliffs, N.J., 1985).



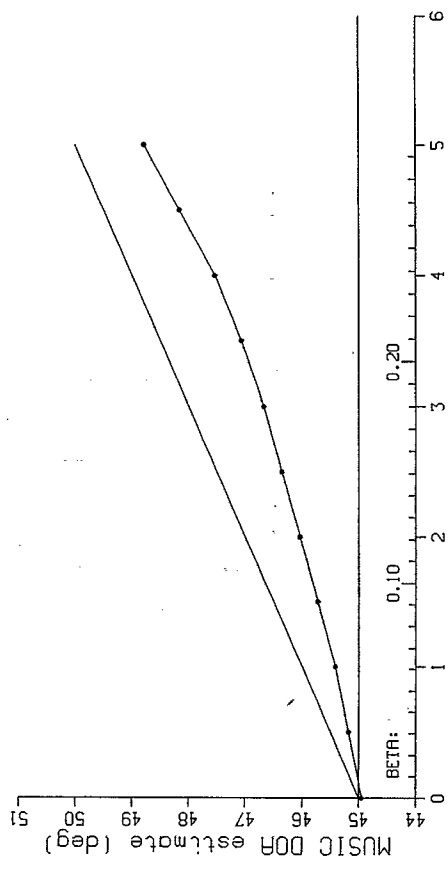
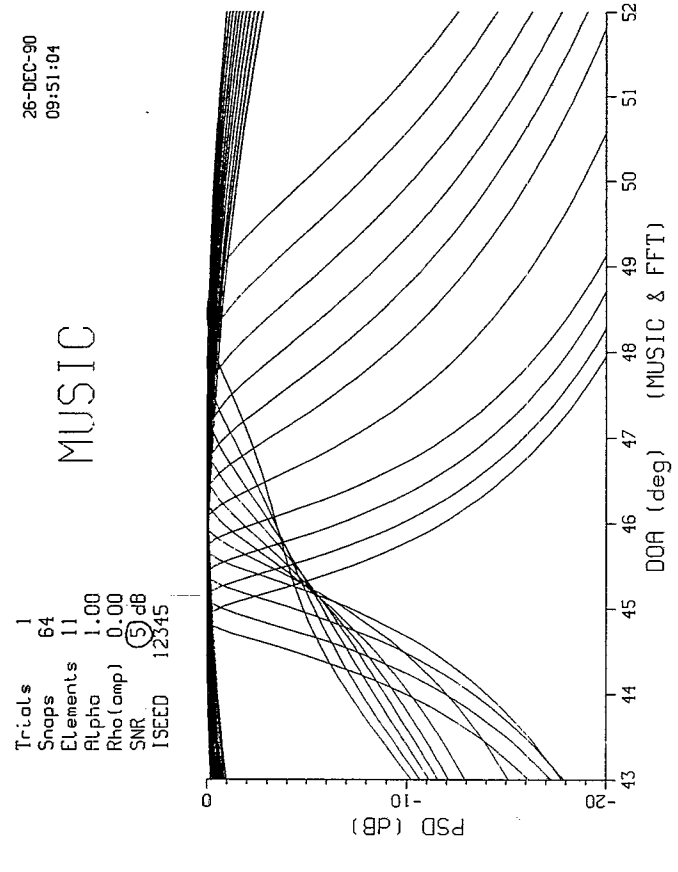


Fig. A1.2 - MUSIC algorithm; perfect decorrelation; 64 snapshots; 7 dB SNR



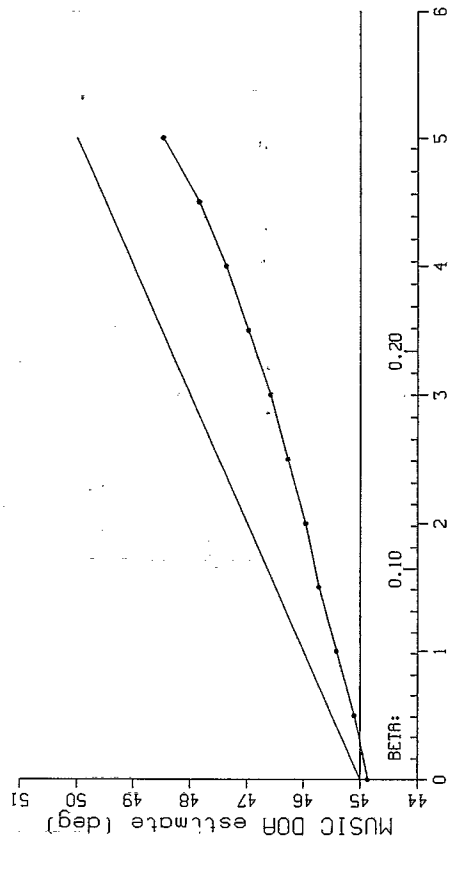
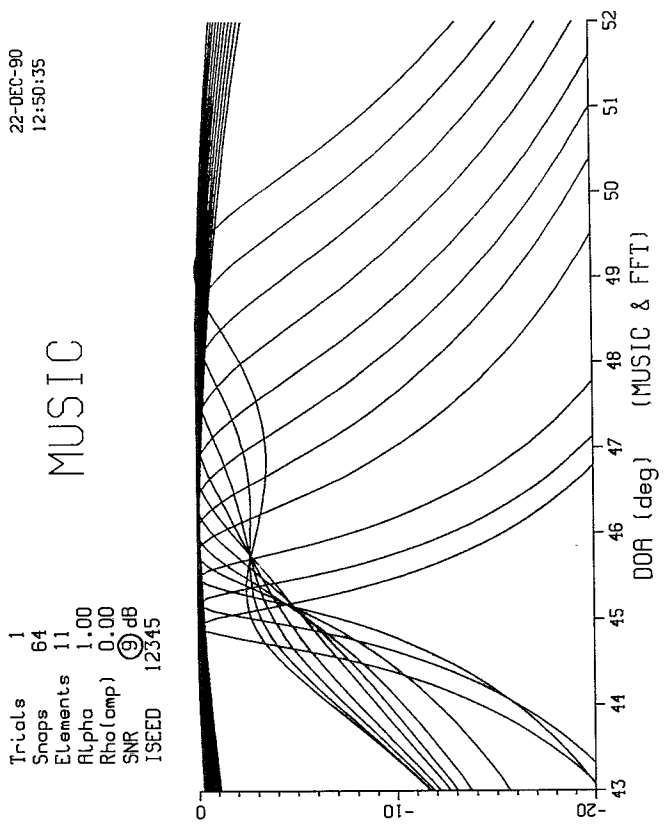


Fig. A1.1 - MUSIC algorithm; perfect decorrelation; 64 snapshots; 5 dB SNR



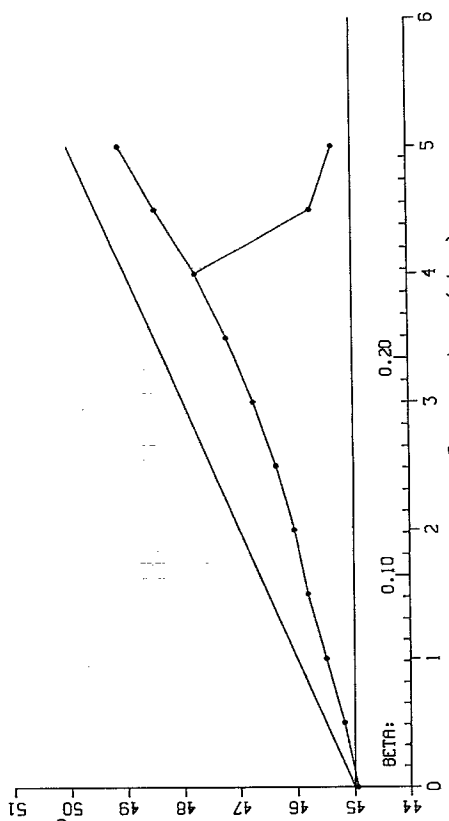
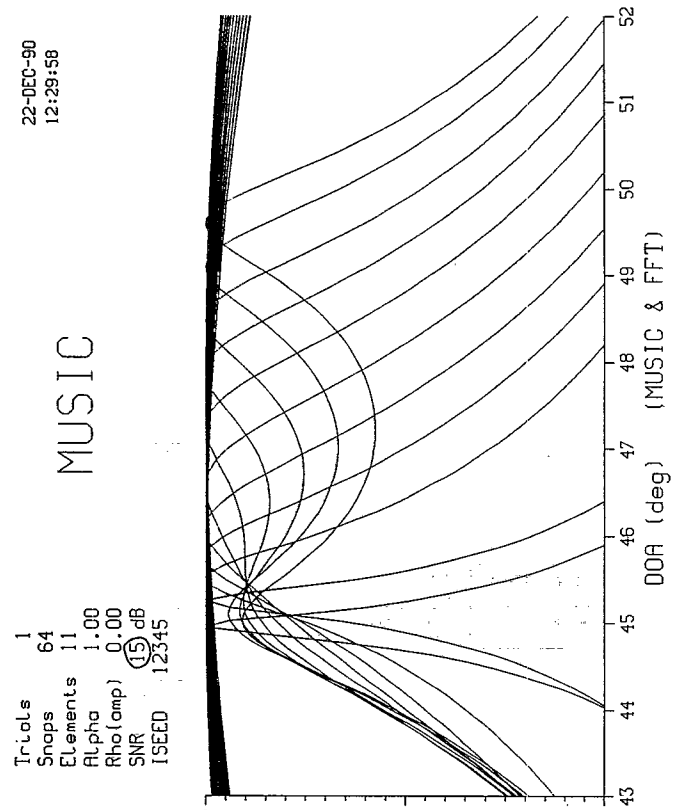


Fig. A1.3 - MUSIC algorithm; perfect decorrelation; 64 snapshots; 8 dB SNR

Fig. A1.4 - MUSIC algorithm; perfect decorrelation; 64 snapshots; 9 dB SNR



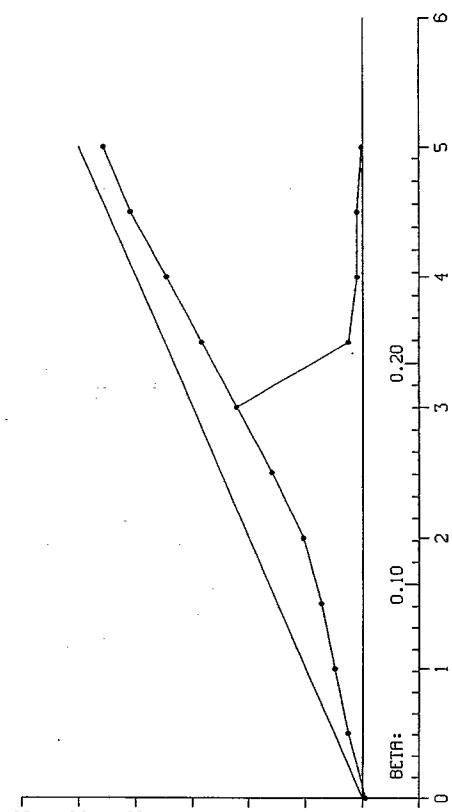
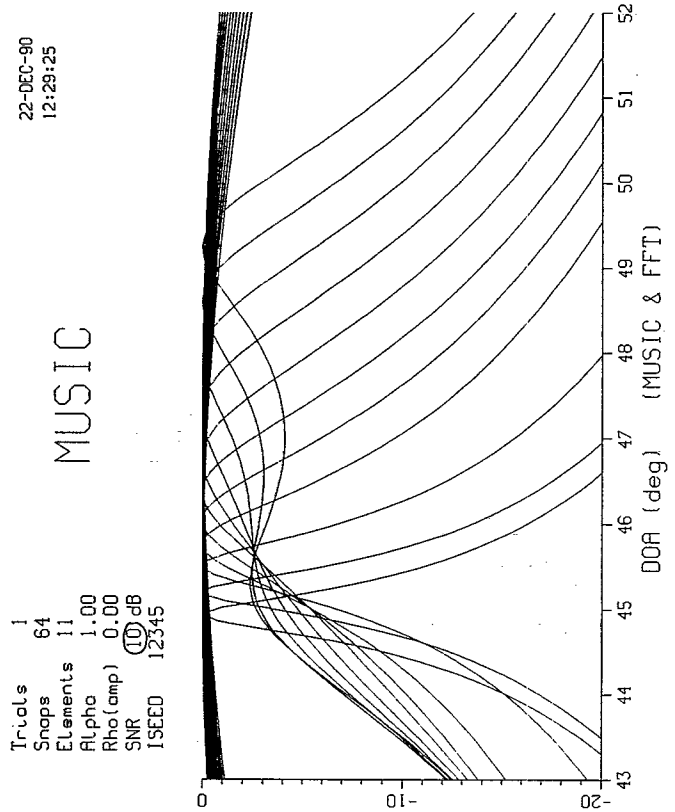


Fig. A1.6 - MUSIC algorithm; perfect decorrelation; 64 snapshots; 15 dB SNR



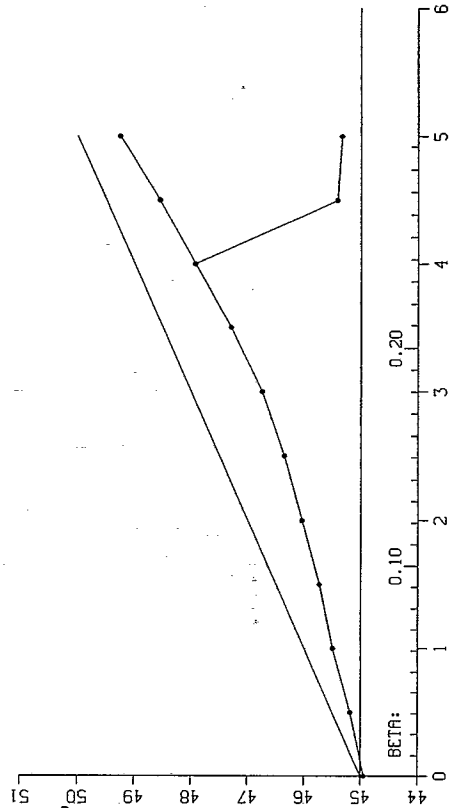
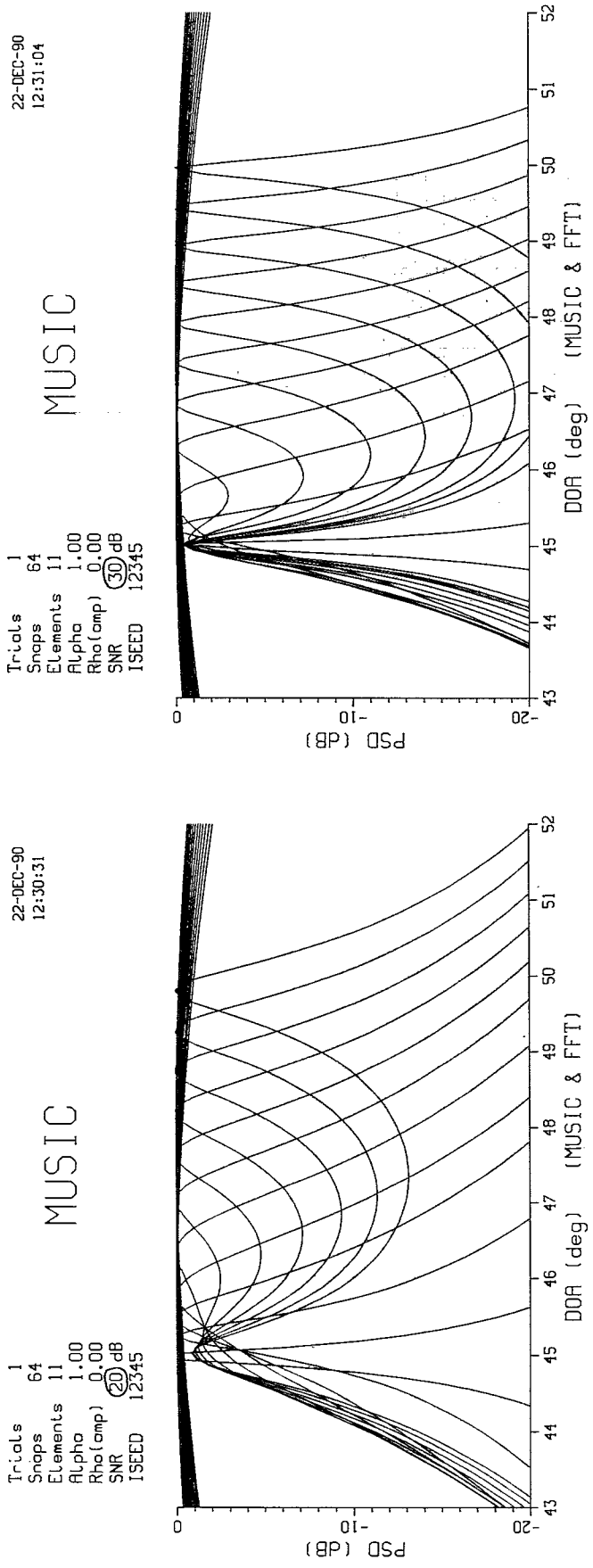


Fig. A1.5 - MUSIC algorithm; perfect decorrelation; 64 snapshots; 10 dB SNR



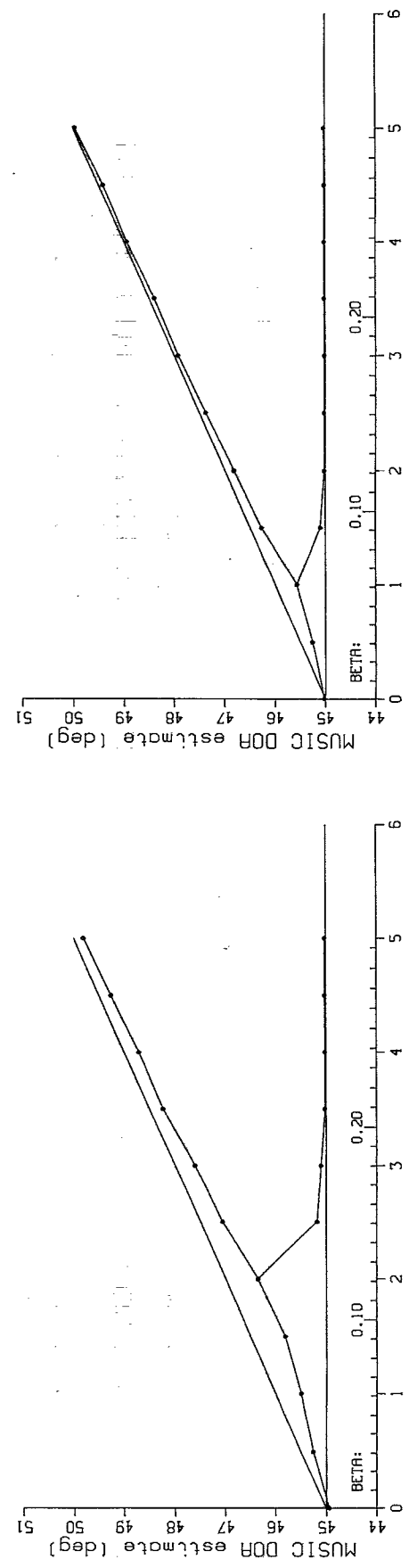
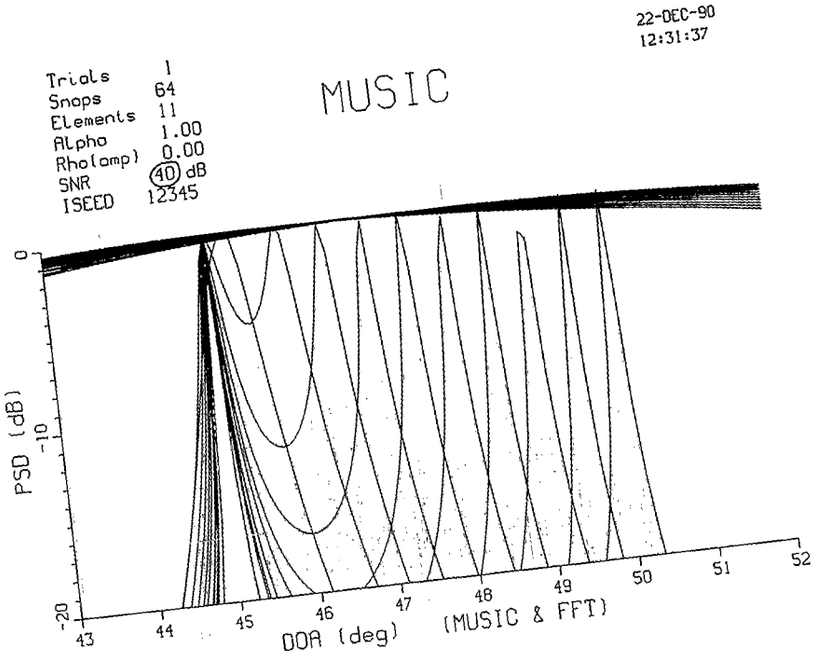


Fig. A1.7 - MUSIC algorithm; perfect decorrelation; 64 snapshots; 20 dB SNR

Fig. A1.8 - MUSIC algorithm; perfect decorrelation; 64 snapshots; 30 dB SNR



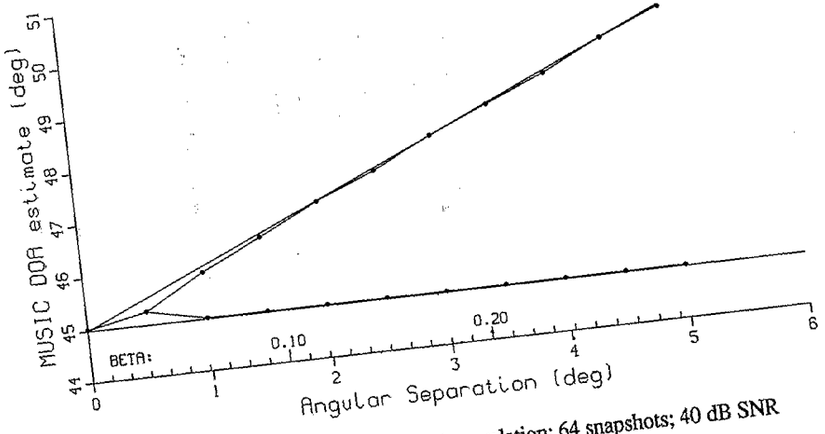
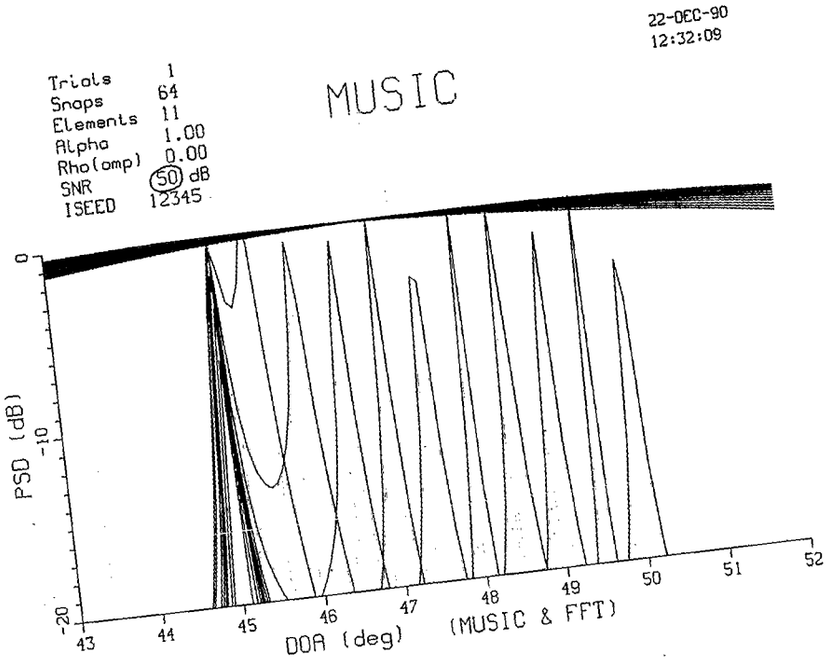


Fig. A1.9 - MUSIC algorithm; perfect decorrelation; 64 snapshots; 40 dB SNR



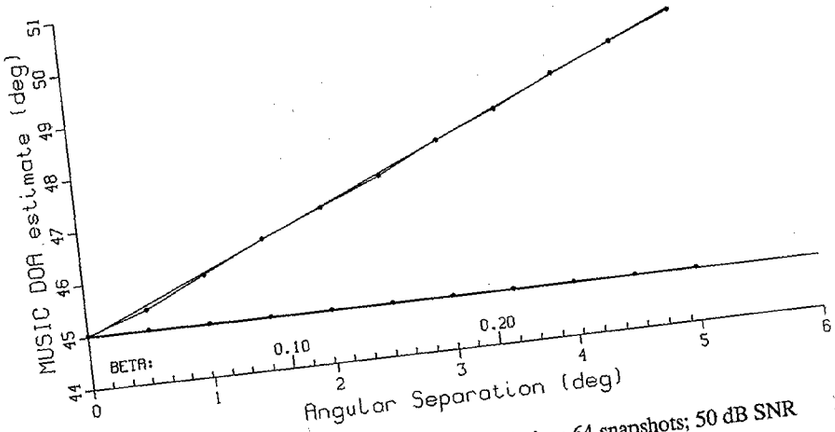


Fig. A1.10 - MUSIC algorithm; perfect decorrelation; 64 snapshots; 50 dB SNR

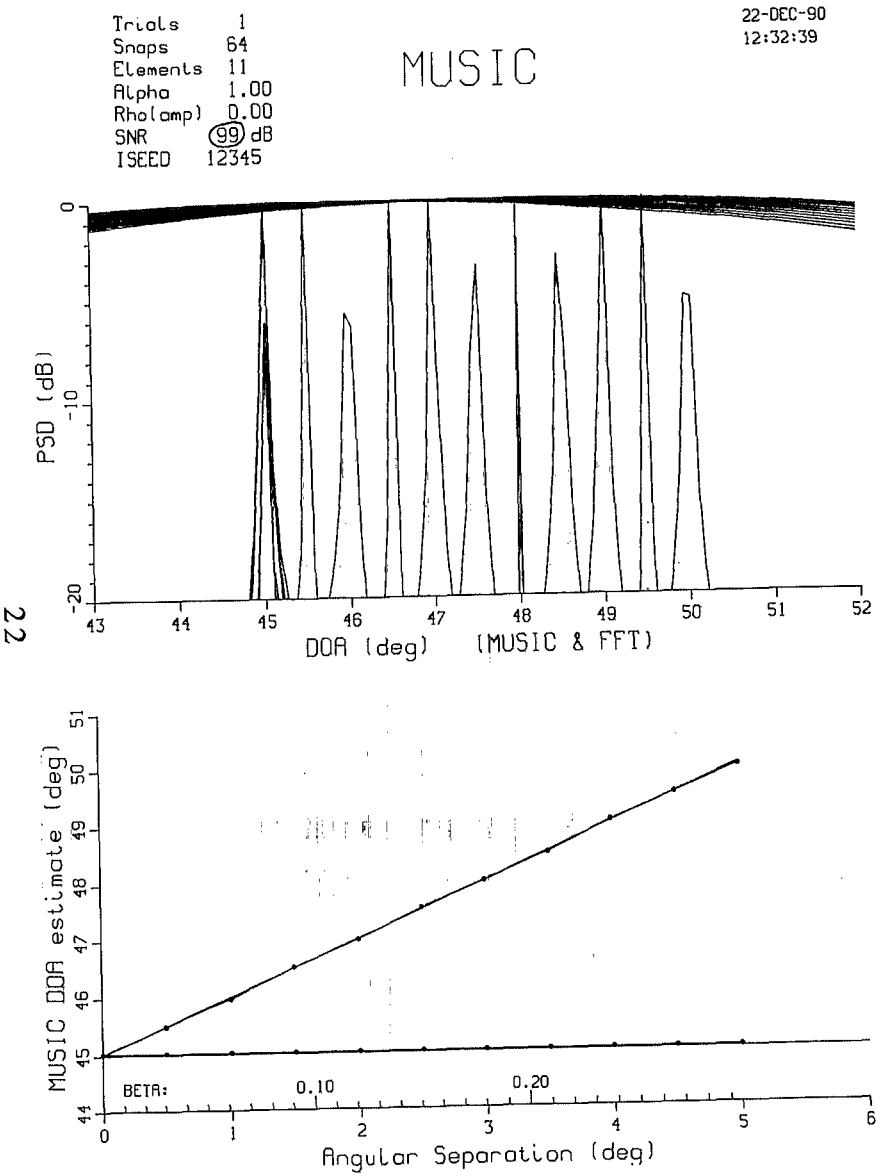


Fig. A1.11 - MUSIC algorithm; pertect decorrelation; 64 snapshots; 99 dB SNR

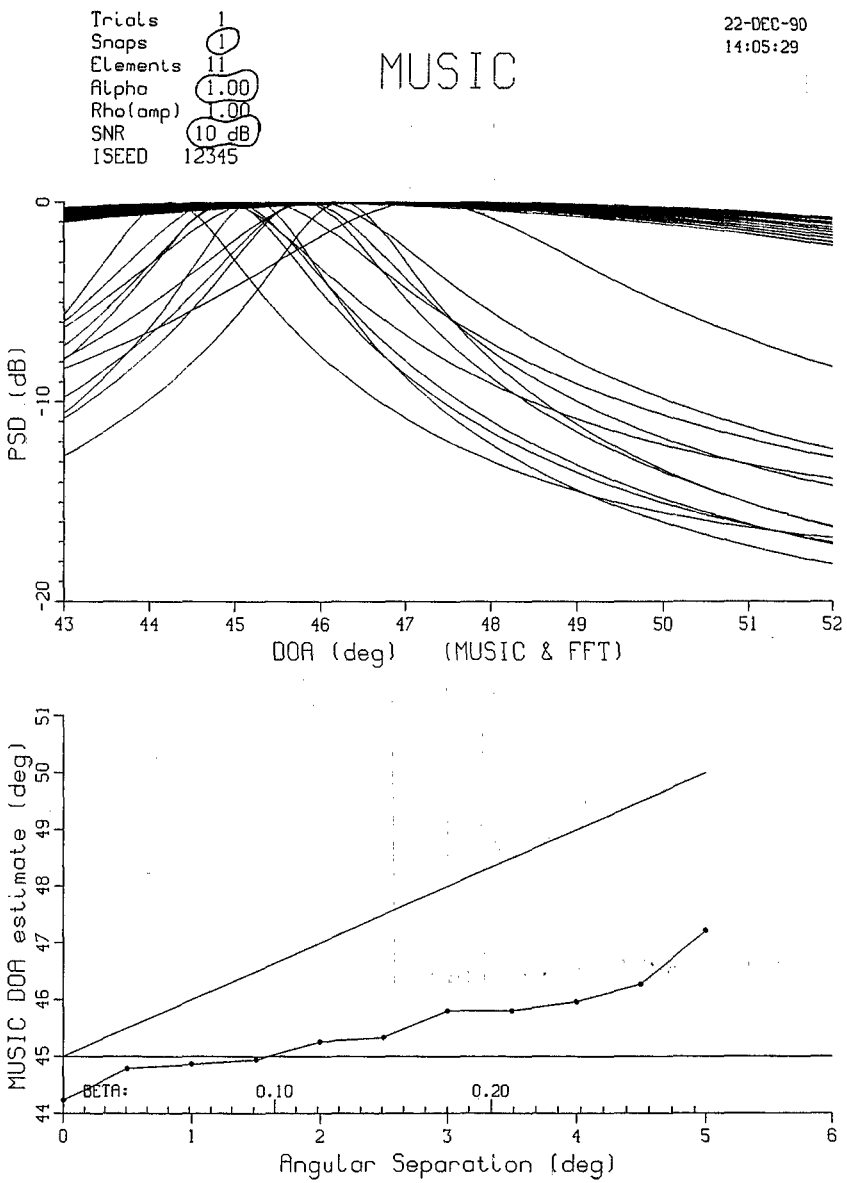


Fig. A2.1 - MUSIC algorithm; perfect decorrelation; 1 snapshot; 10 dB SNR

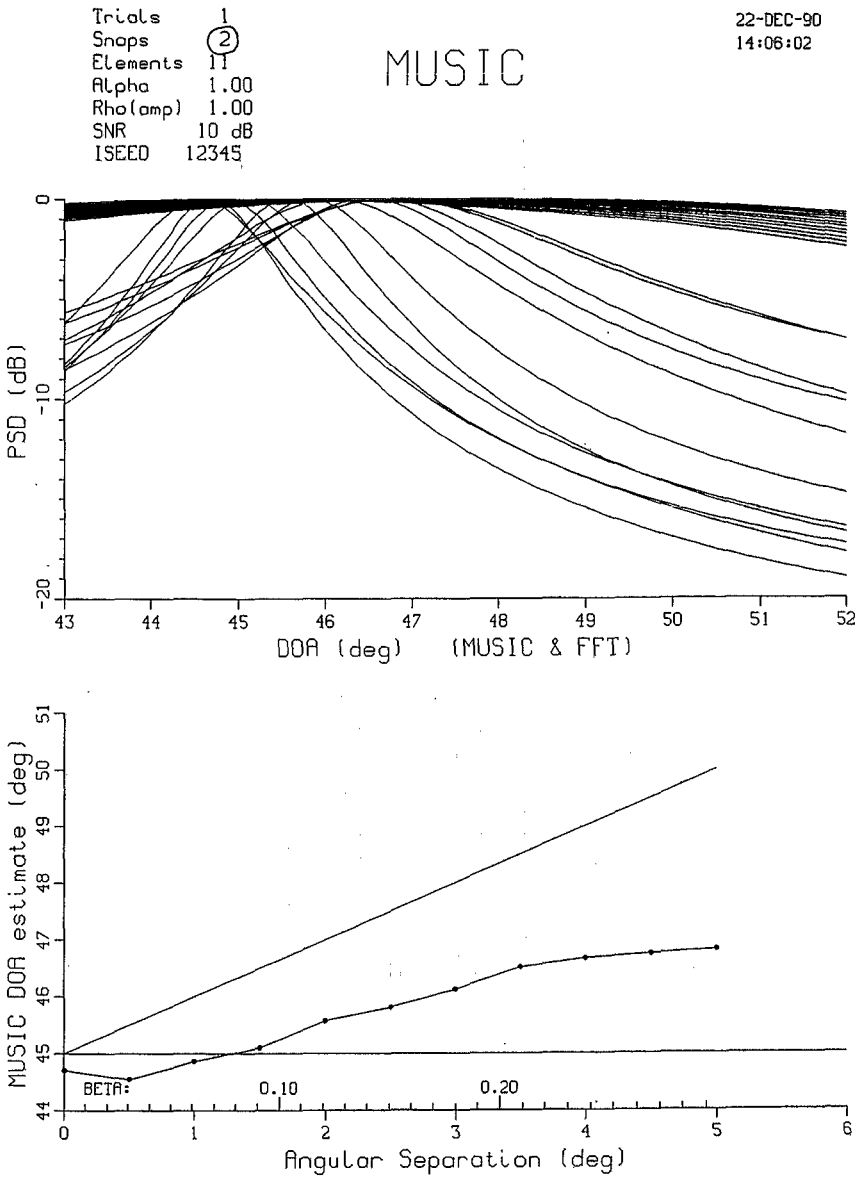


Fig. A2.2 - MUSIC algorithm; perfect decorrelation; 2 snapshots; 10 dB SNR

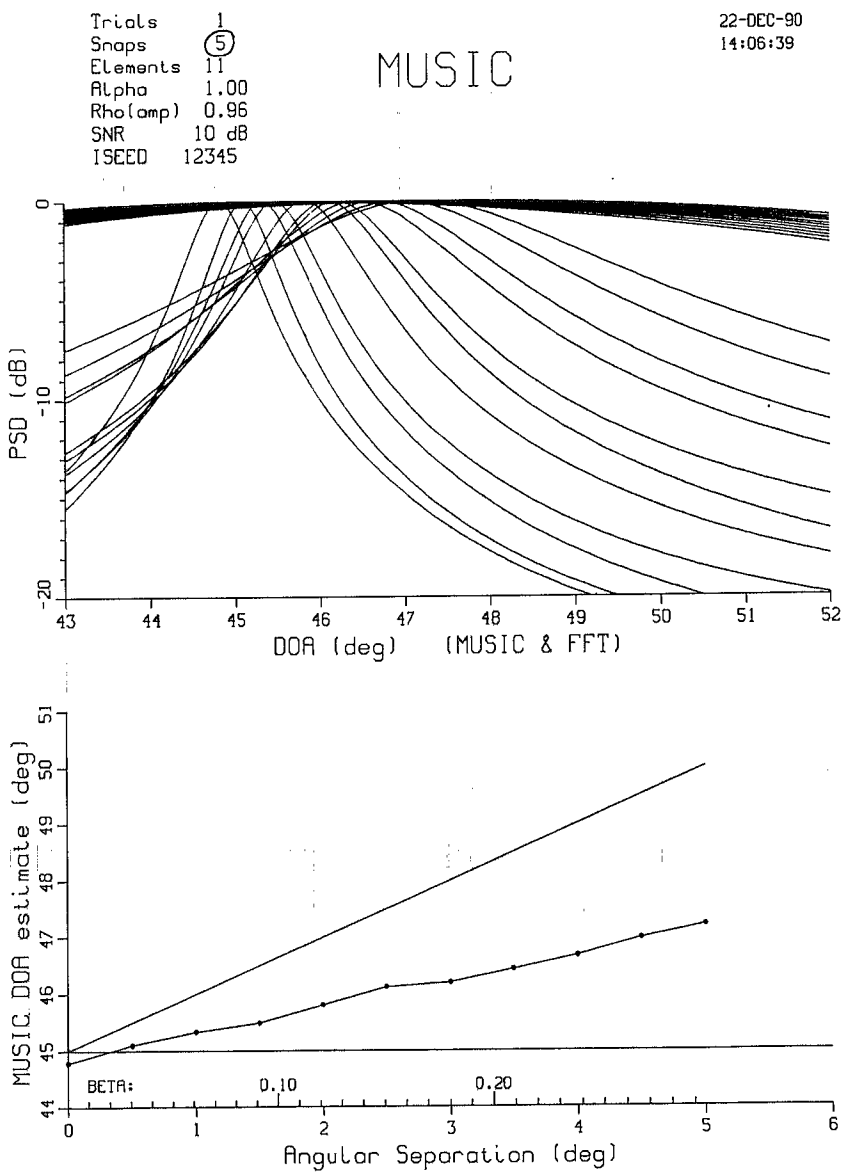


Fig. A2.3 - MUSIC algorithm; perfect decorrelation; 5 snapshots; 10 dB SNR

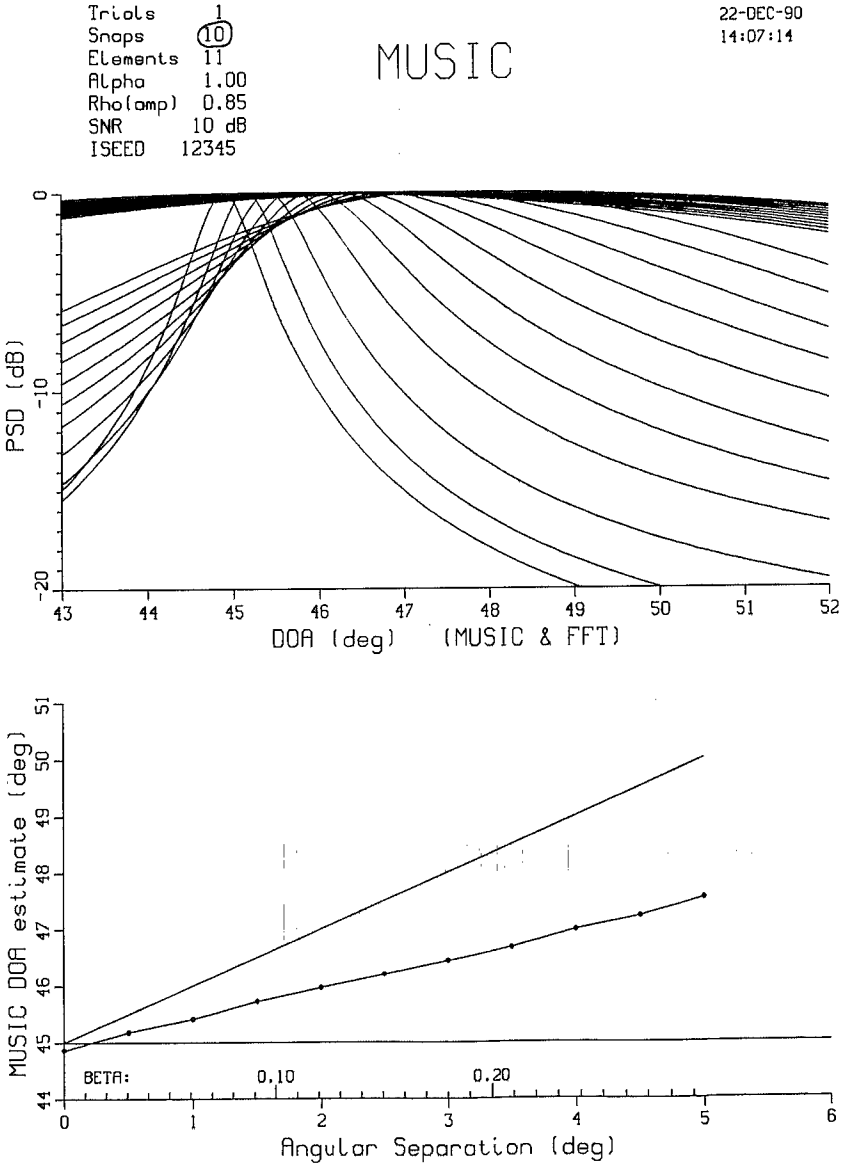
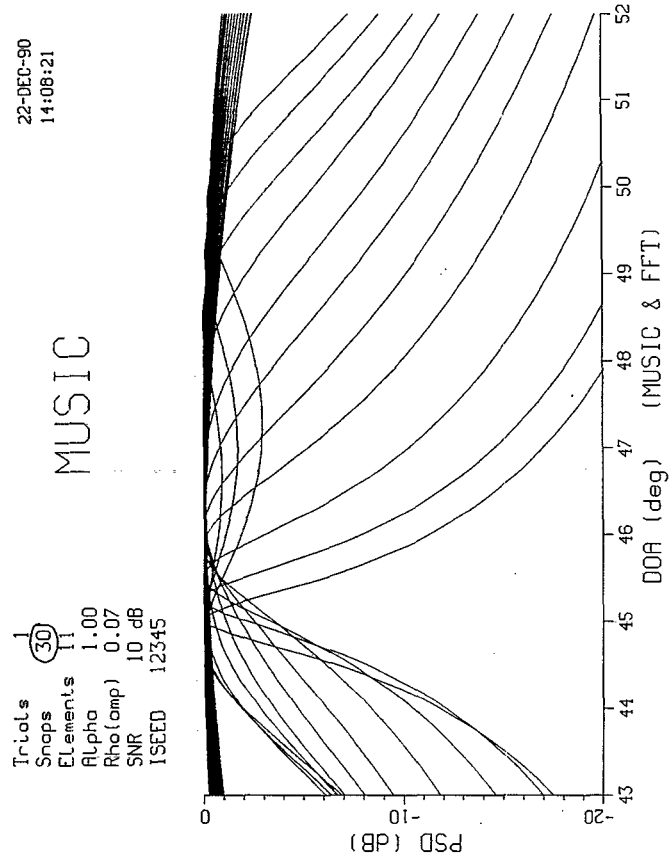


Fig. A2.4 - MUSIC algorithm; perfect decorrelation; 10 snapshots; 10 dB SNR



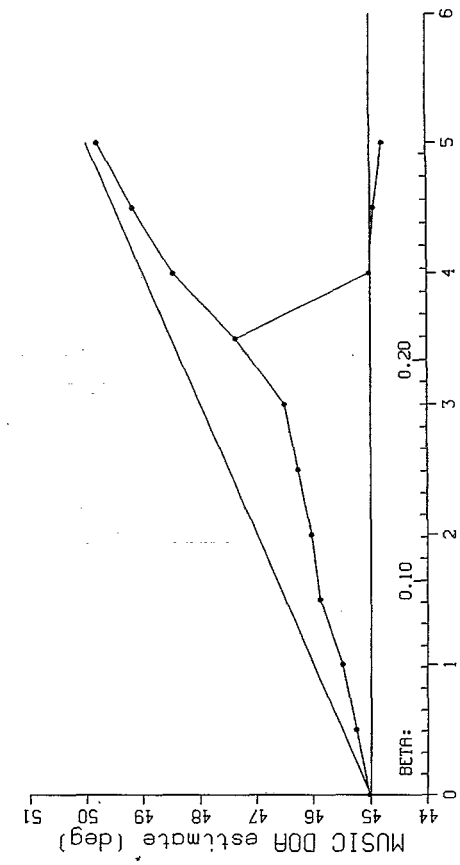
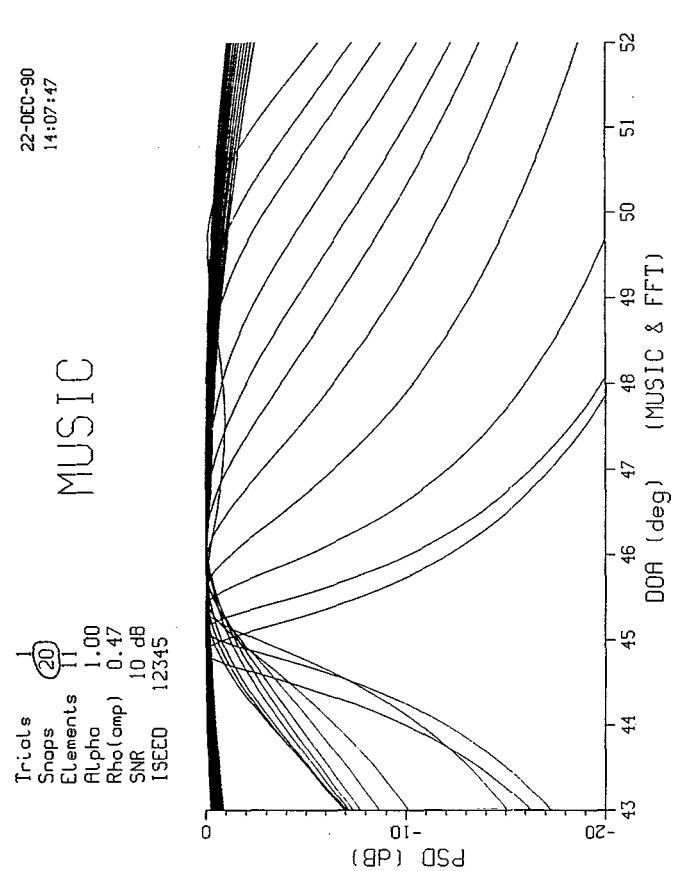


Fig. A2.5 - MUSIC algorithm; perfect decorrelation; 20 snapshots; 10 dB SNR



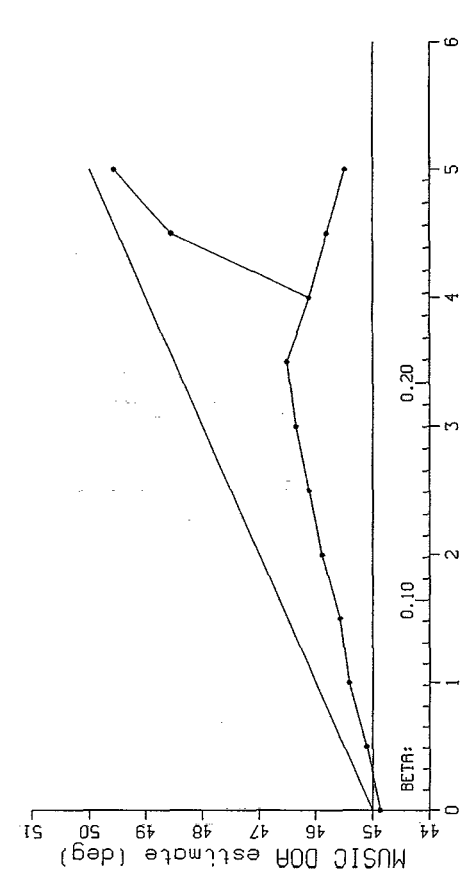
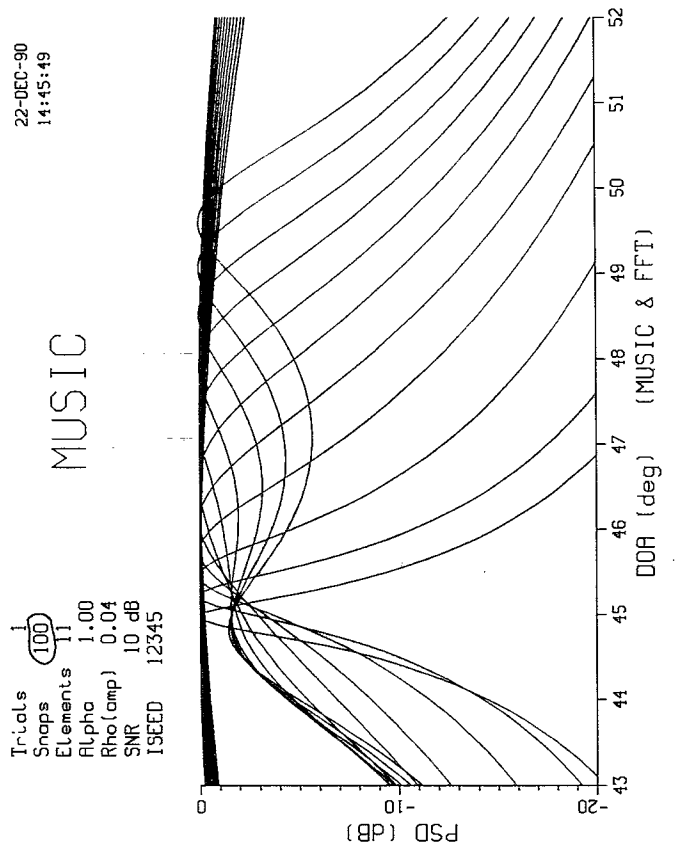


Fig. A2.6 - MUSIC algorithm; perfect decorrelation; 30 snapshots; 10 dB SNR



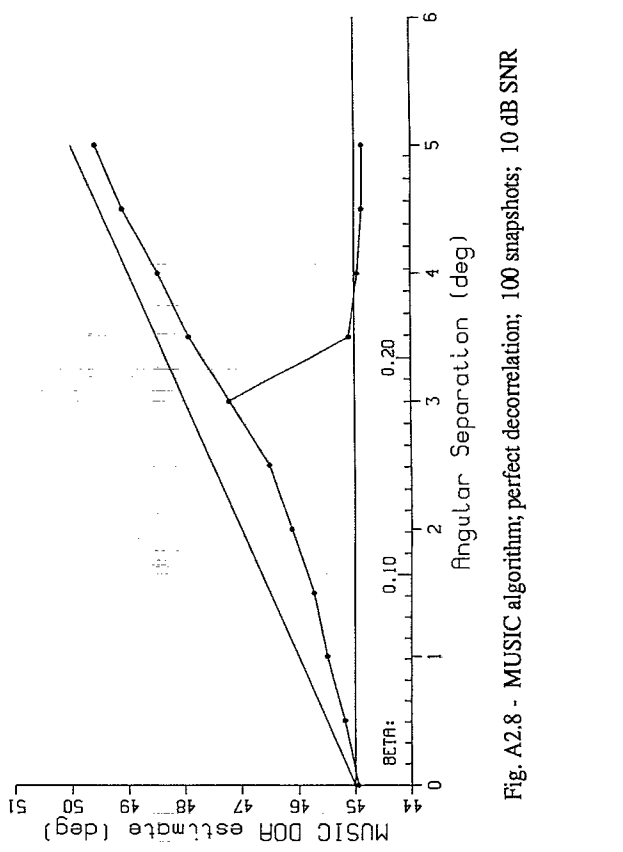


Fig. A2.7 - MUSIC algorithm; perfect decorrelation; 64 snapshots; 10 dB SNR

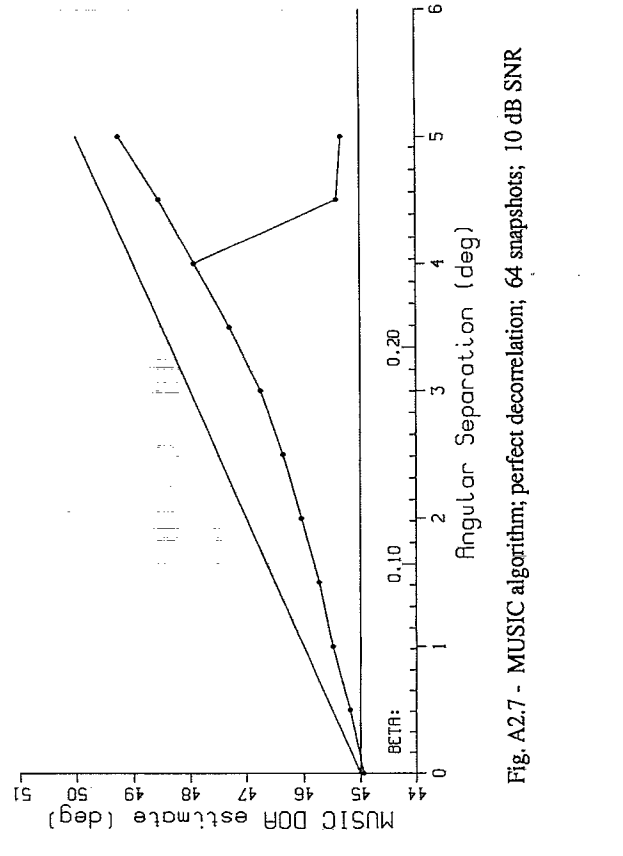


Fig. A2.8 - MUSIC algorithm; perfect decorrelation; 100 snapshots; 10 dB SNR

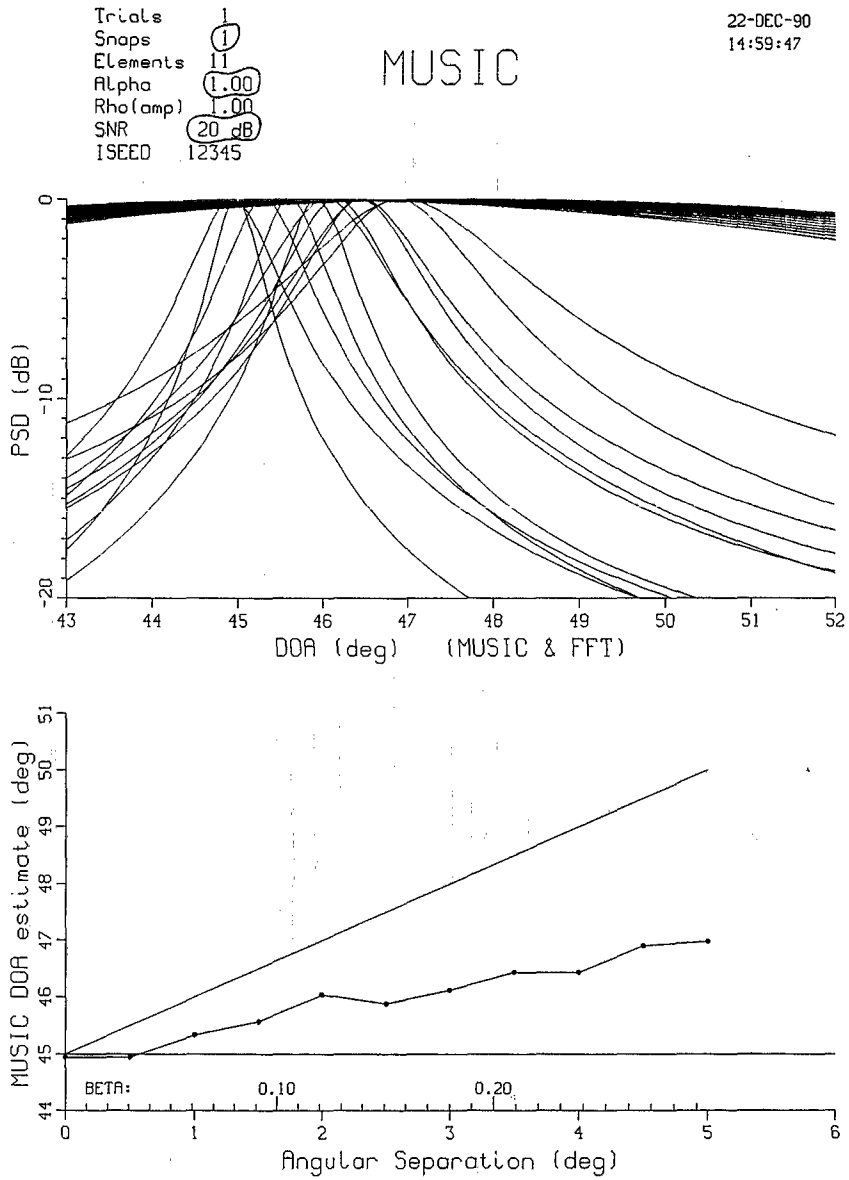


Fig. A3.1 - MUSIC algorithm; perfect decorrelation; 1 snapshot; 20 dB SNR

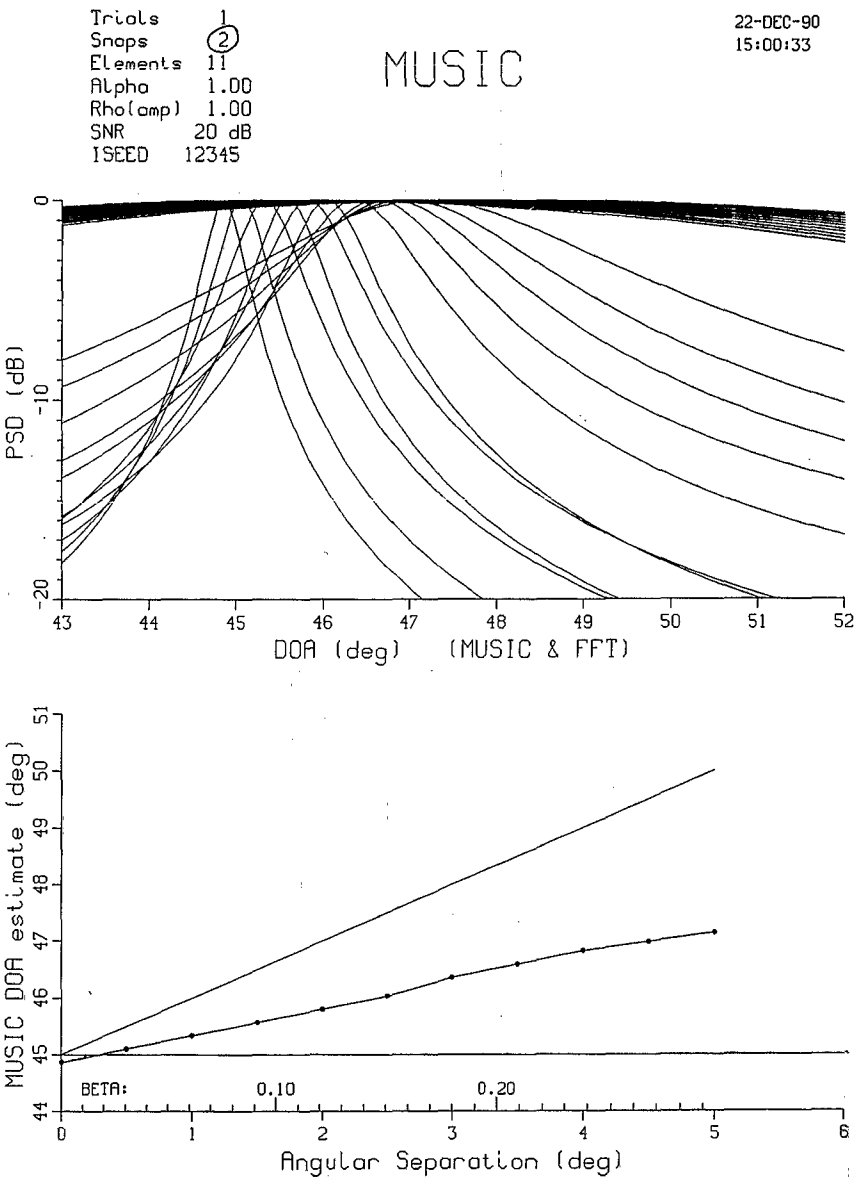


Fig. A3.2 - MUSIC algorithm; perfect decorrelation; 2 snapshots; 20 dB SNR

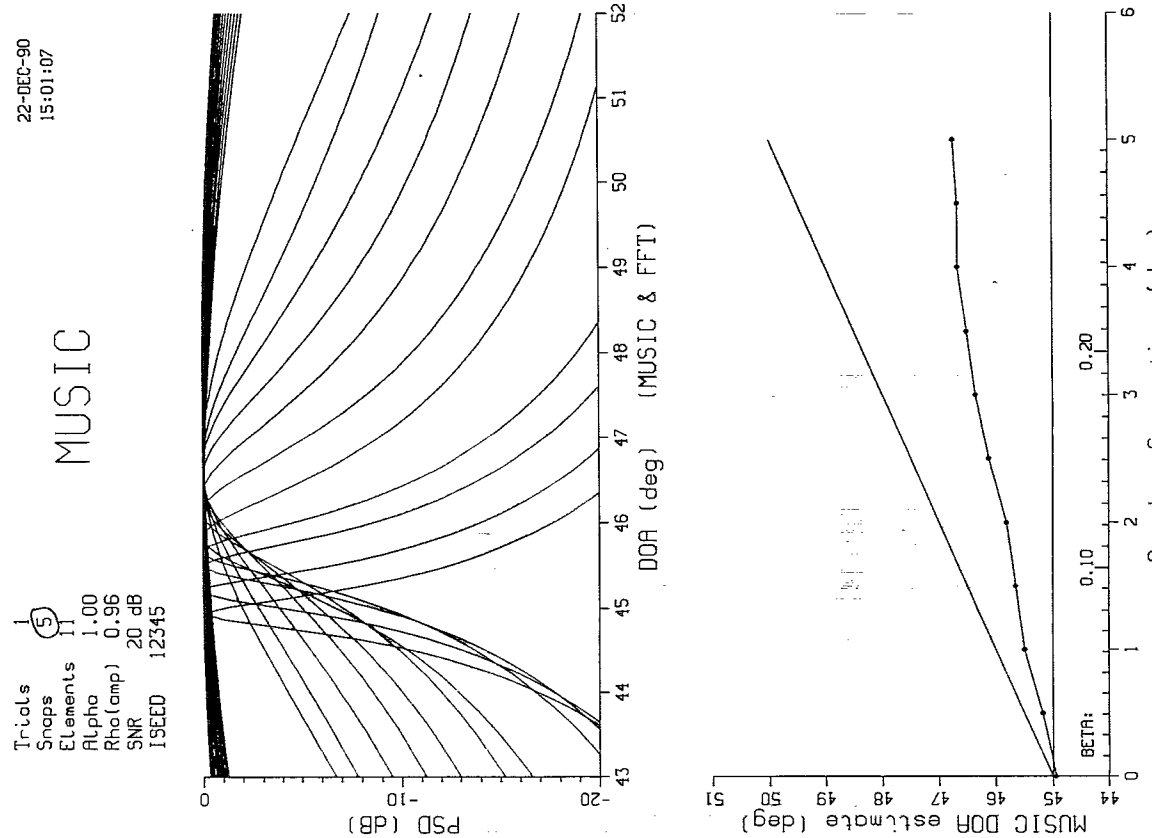


Fig. A3.3 - MUSIC algorithm; perfect decorrelation; 5 snapshots; 20 dB SNR

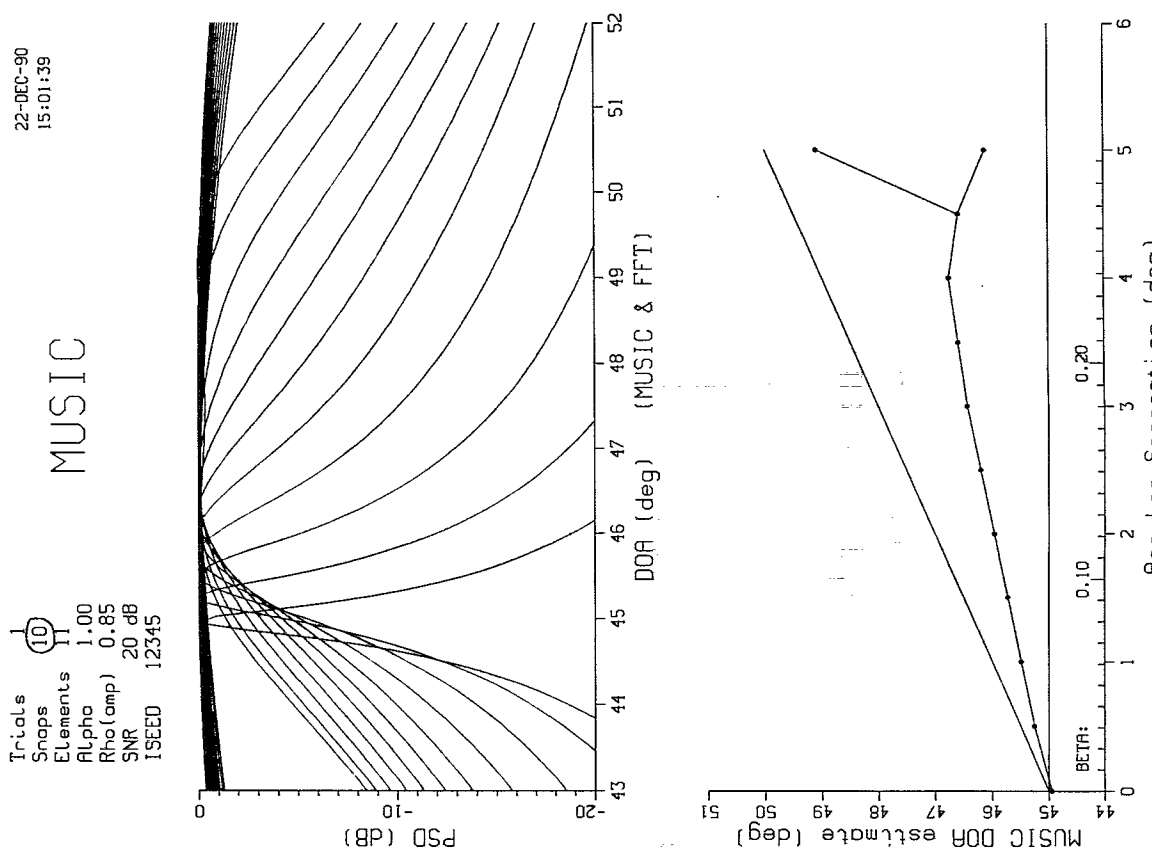


Fig. A3.4 - MUSIC algorithm; perfect decorrelation; 10 snapshots; 20 dB SNR

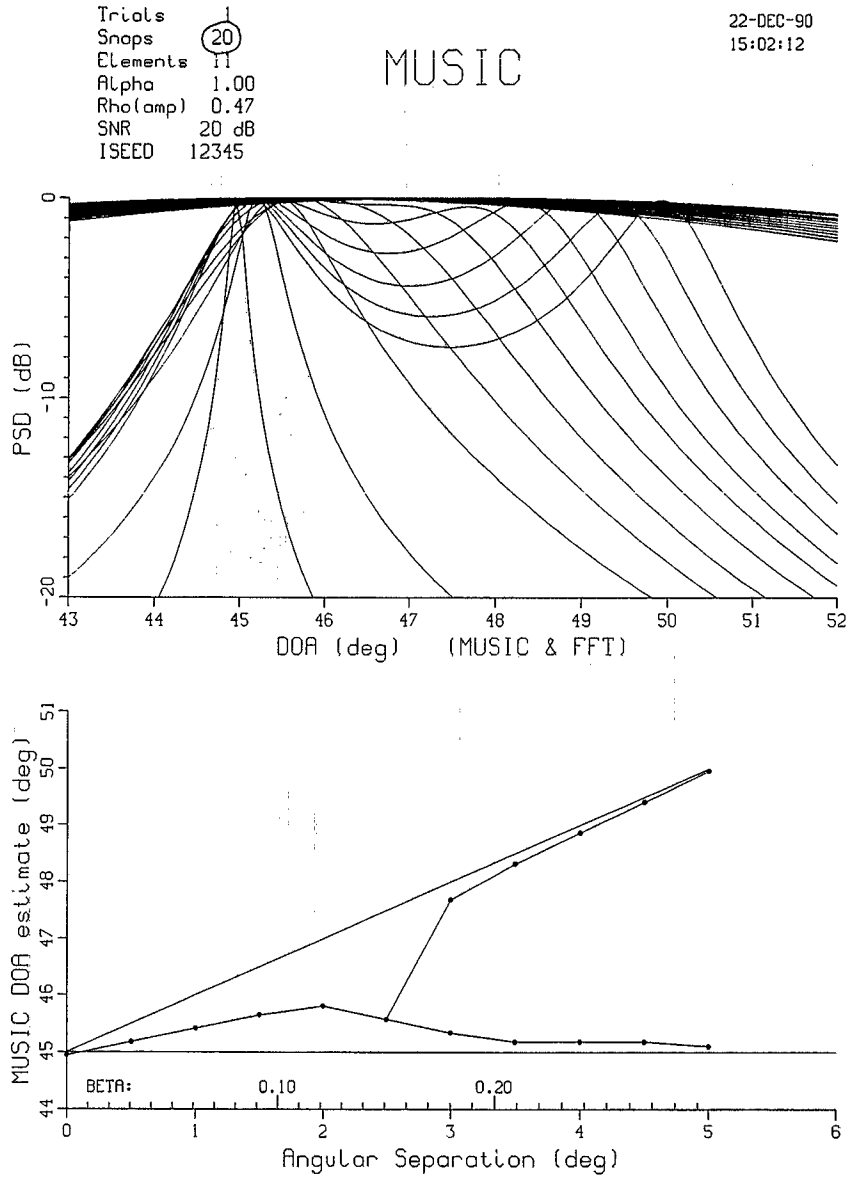


Fig. A3.5 - MUSIC algorithm; perfect decorrelation; 20 snapshots; 20 dB SNR

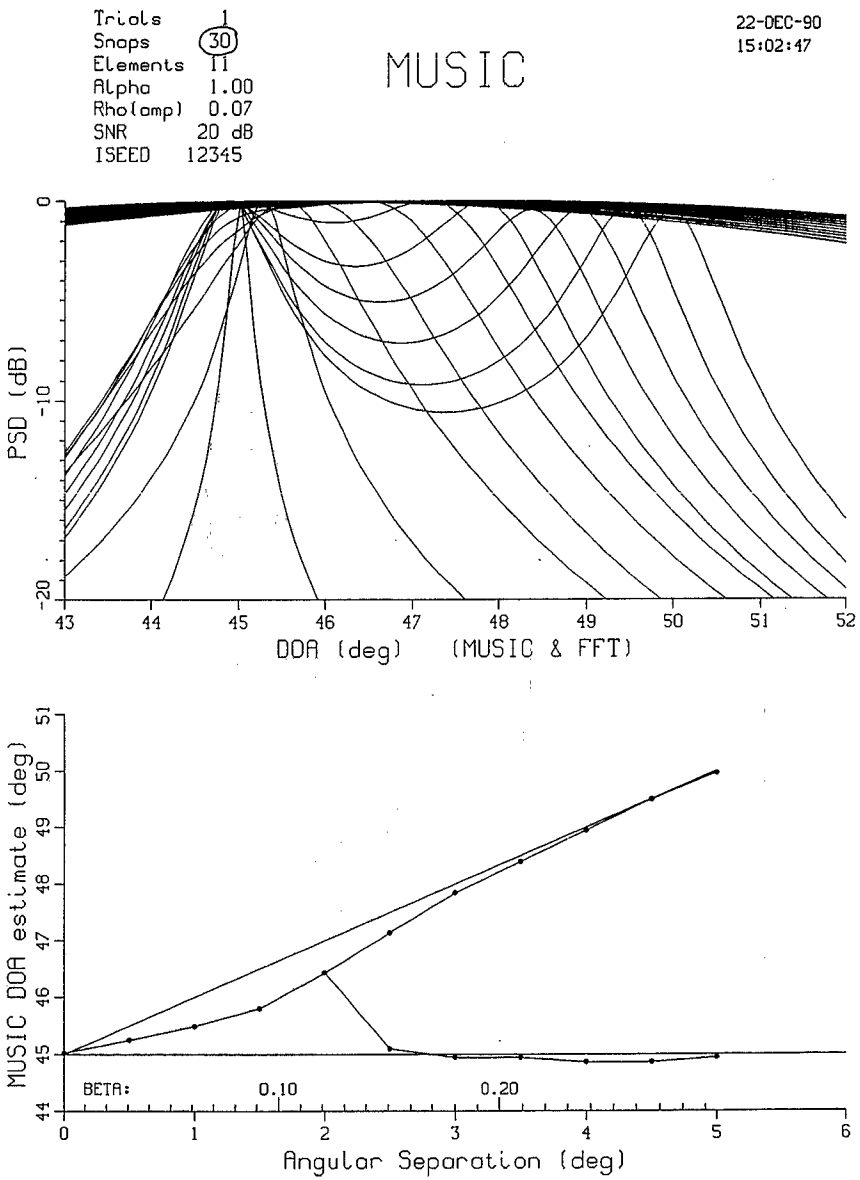
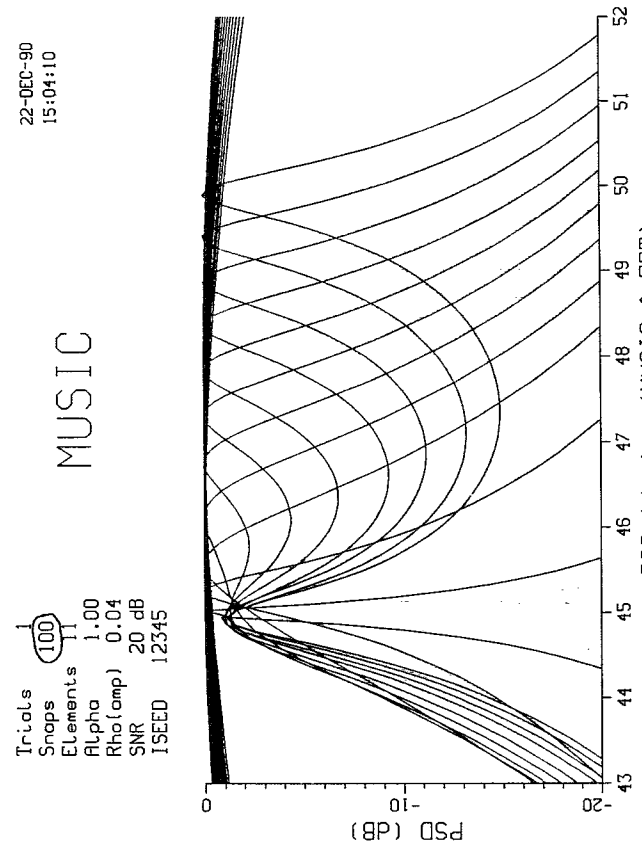


Fig. A3.6 - MUSIC algorithm; perfect decorrelation; 30 snapshots; 20 dB SNR



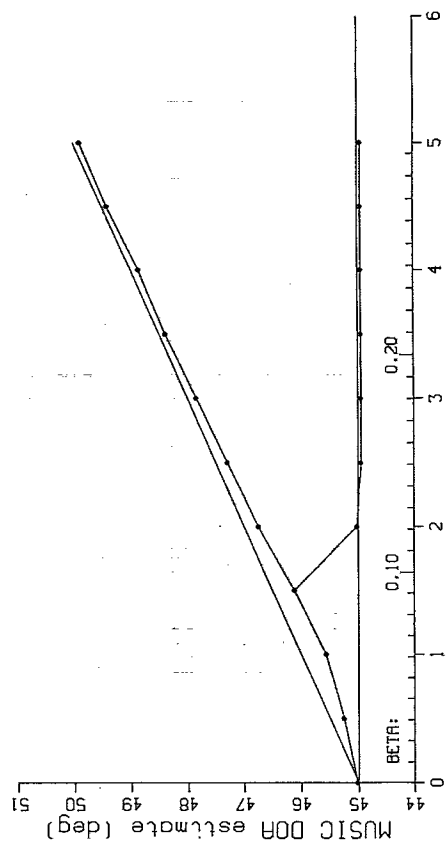


Fig. A3.7 - MUSIC algorithm; perfect decorrelation; 64 snapshots; 20 dB SNR

Fig. A3.8 - MUSIC algorithm; perfect decorrelation; 100 snapshots; 20 dB SNR

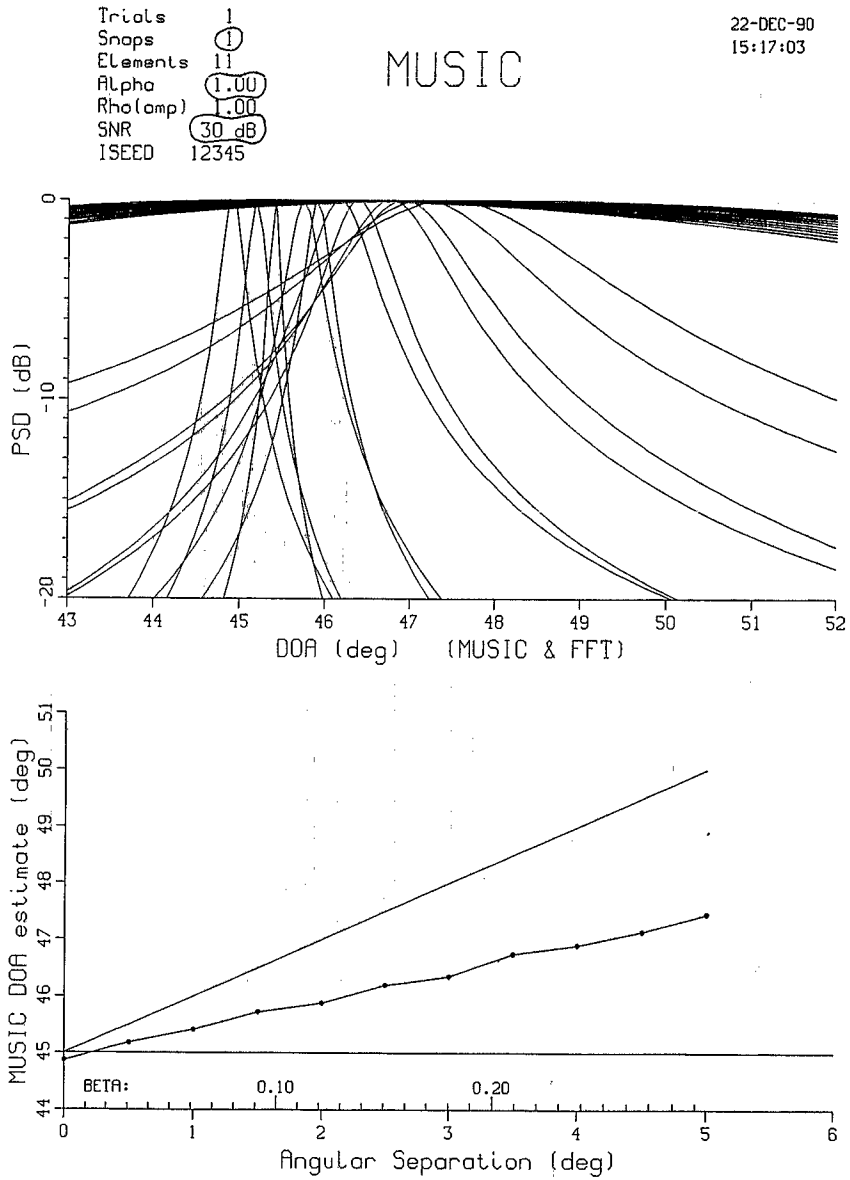


Fig. A4.1 - MUSIC algorithm; perfect decorrelation; 1 snapshot; 30 dB SNR

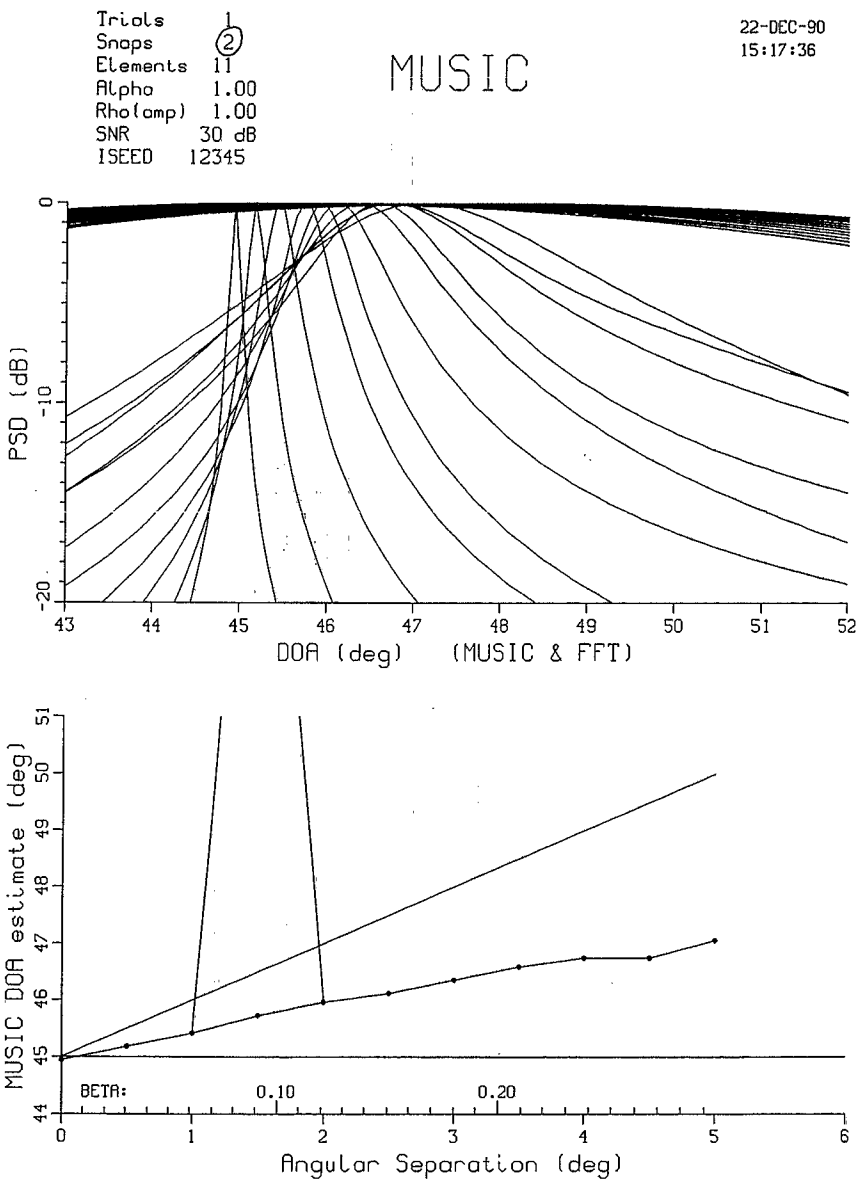
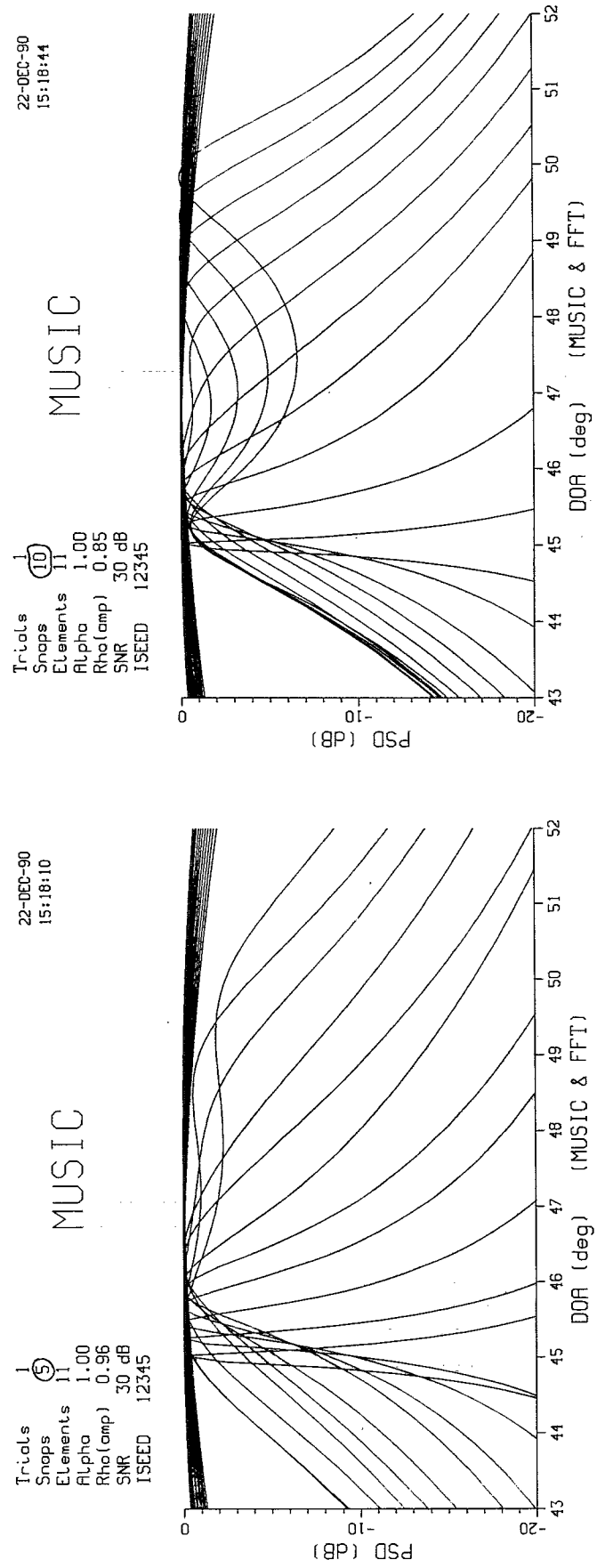


Fig. A4.2 - MUSIC algorithm; perfect decorrelation; 2 snapshots; 30 dB SNR



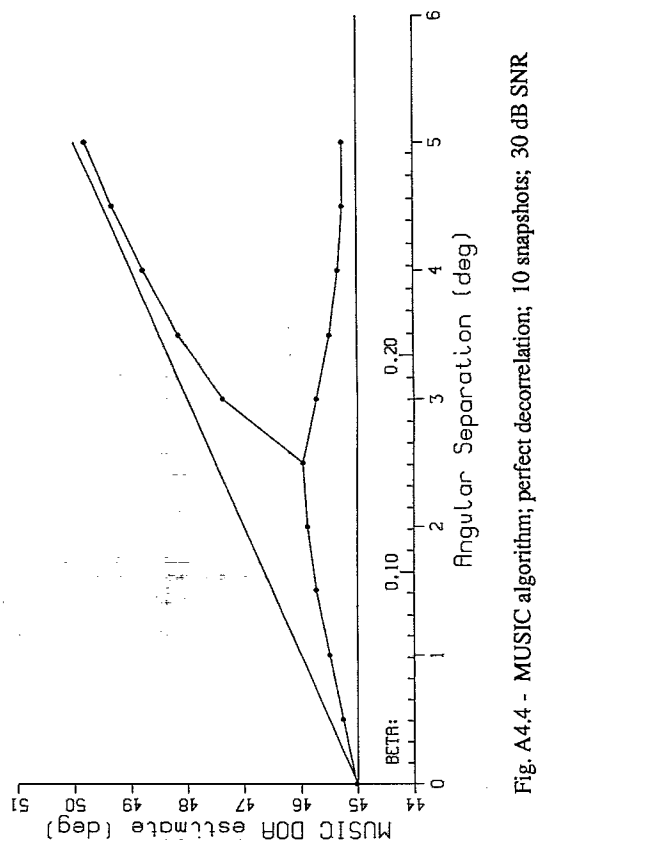


Fig. A4.3 - MUSIC algorithm; perfect decorrelation; 5 snapshots; 30 dB SNR

Fig. A4.4 - MUSIC algorithm; perfect decorrelation; 10 snapshots; 30 dB SNR

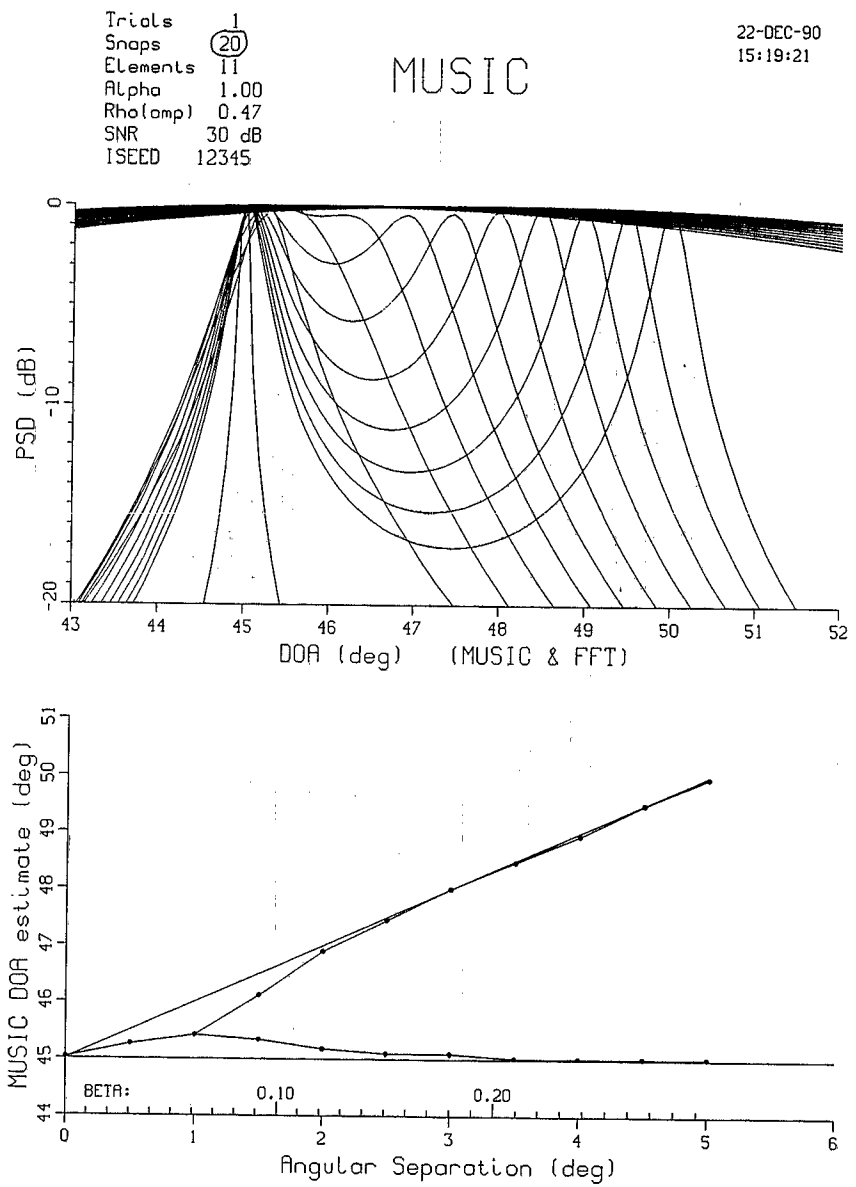


Fig. A4.5 - MUSIC algorithm; perfect decorrelation; 20 snapshots; 30 dB SNR

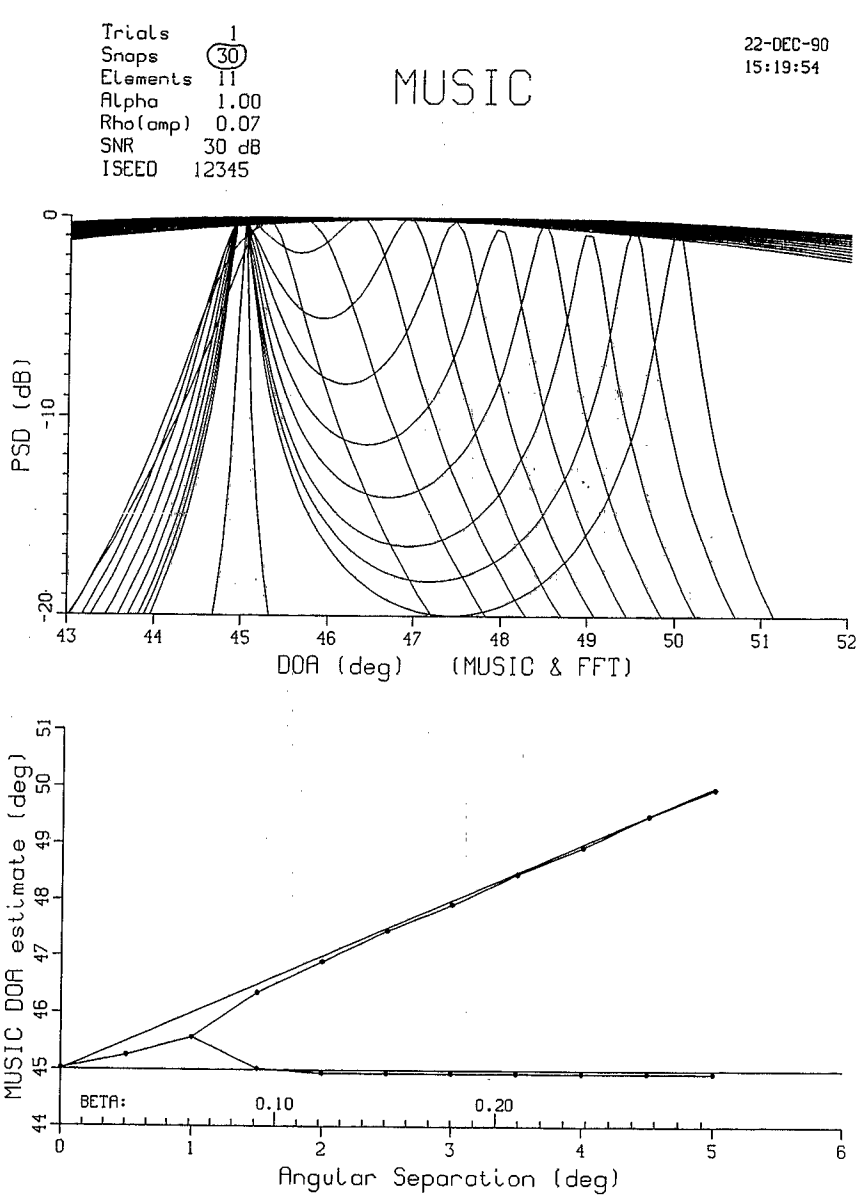


Fig. A4.6 - MUSIC algorithm; perfect decorrelation; 30 snapshots; 30 dB SNR

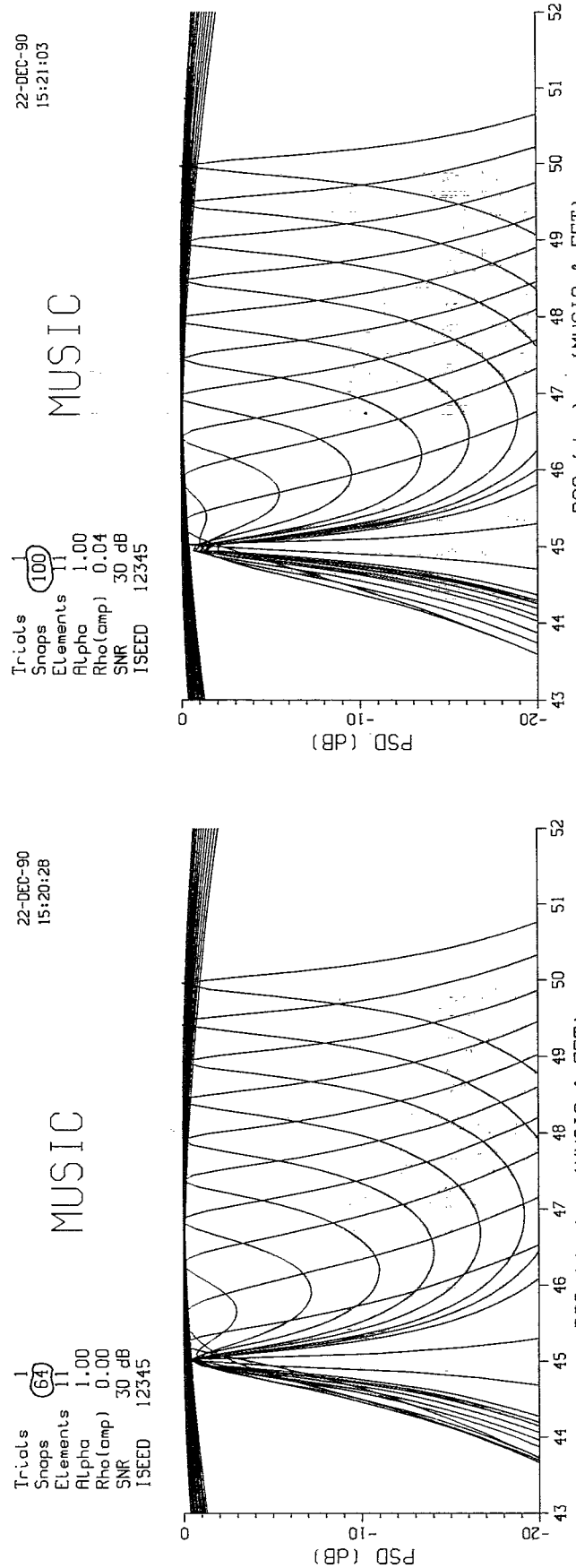


Fig. A4.7 - MUSIC algorithm; perfect decorrelation; 64 snapshots; 30 dB SNR

Fig. A4.8 - MUSIC algorithm; perfect decorrelation; 100 snapshots; 30 dB SNR

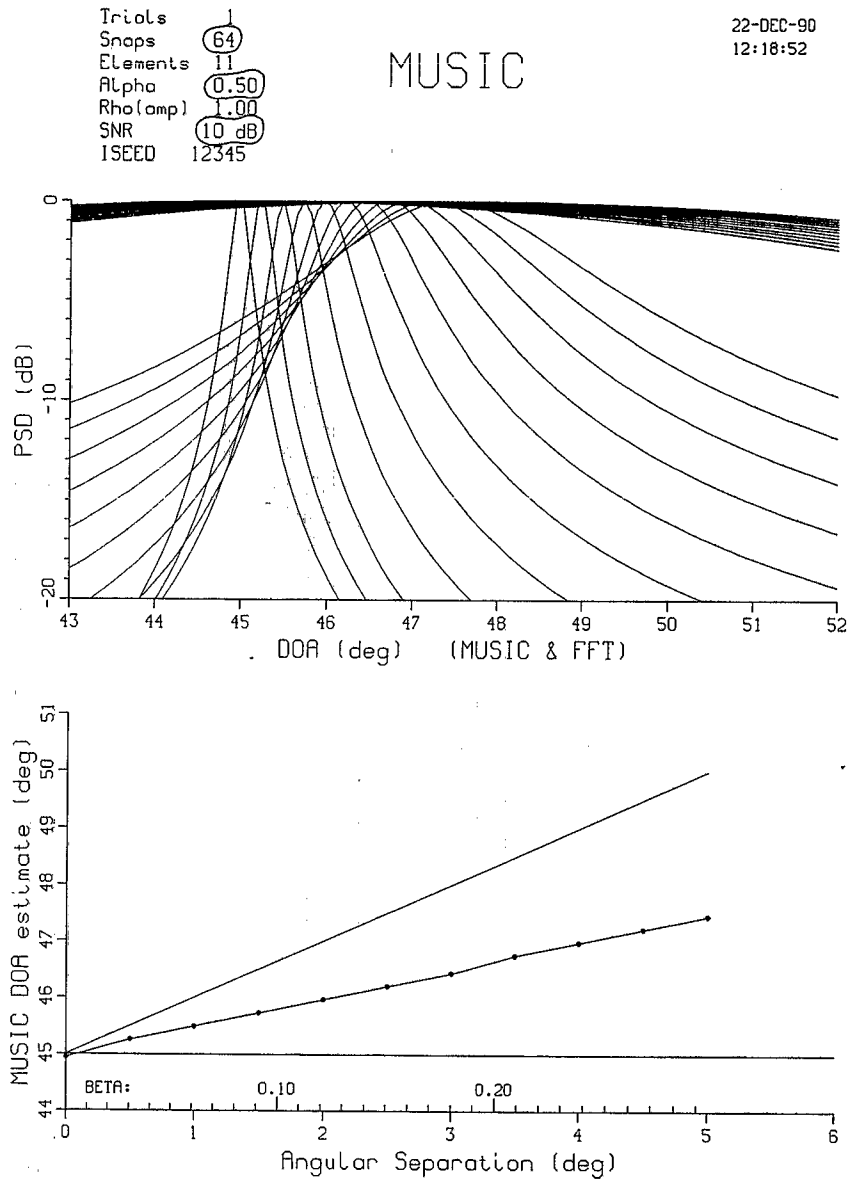


Fig. B1.1 - MUSIC algorithm; high correlation (α=0.50); 64 snapshots; 10 dB SNR

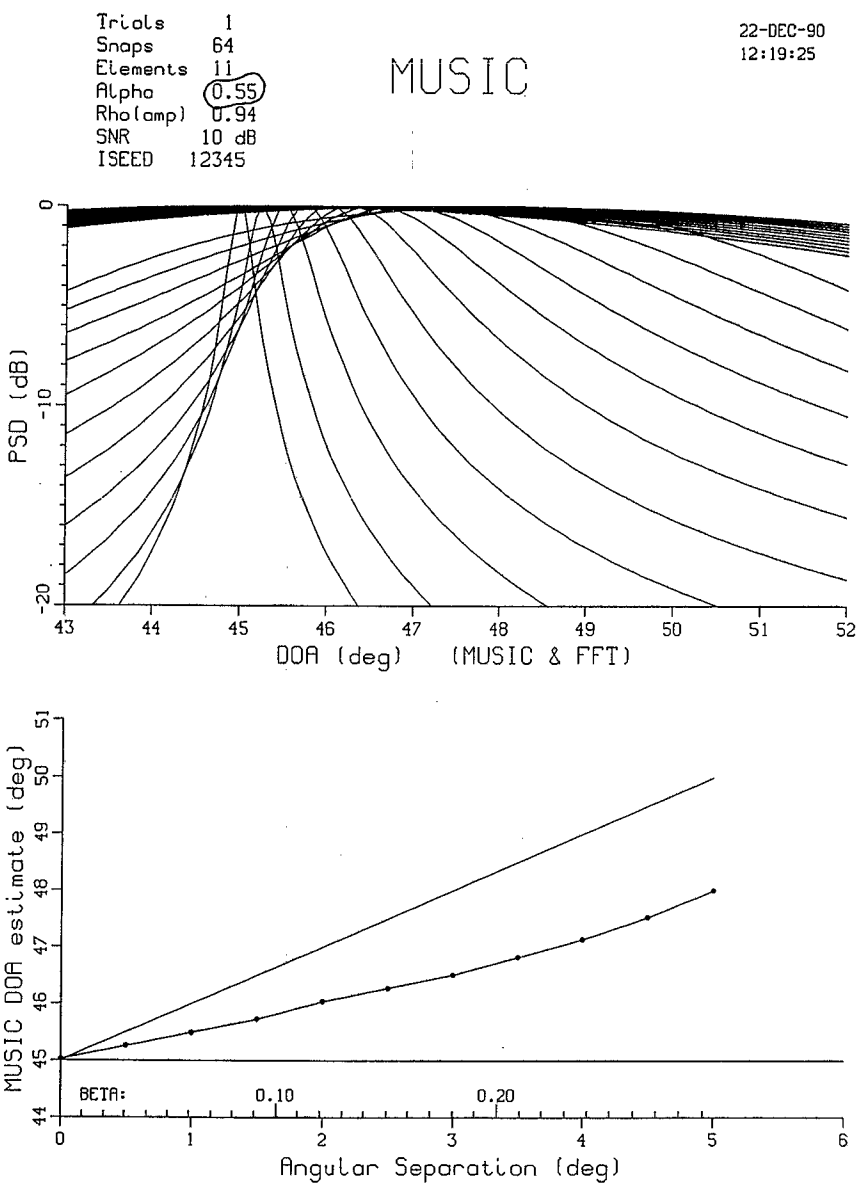
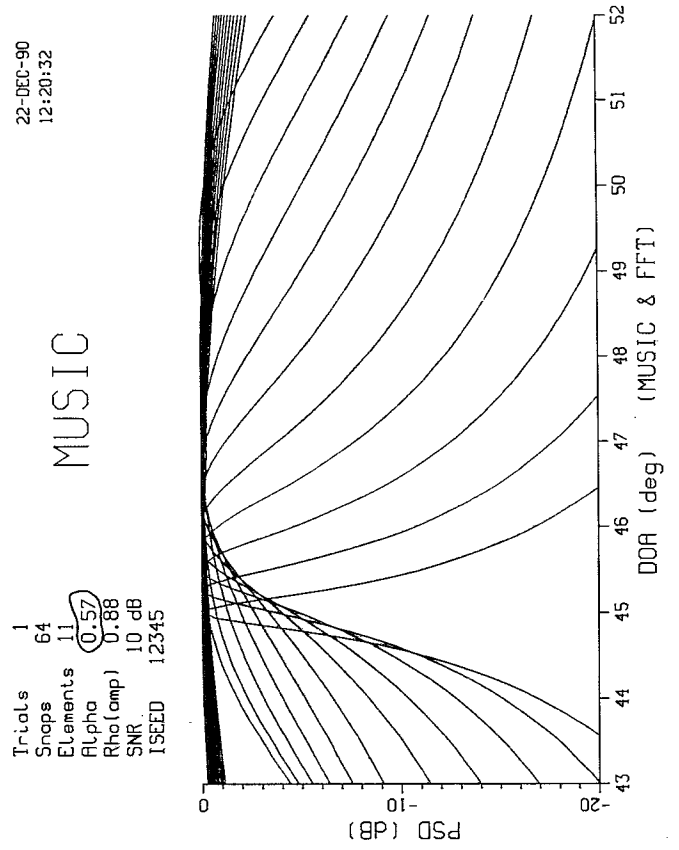


Fig. B1.2 - MUSIC algorithm; high correlation (α=0.55); 64 snapshots; 10 dB SNR



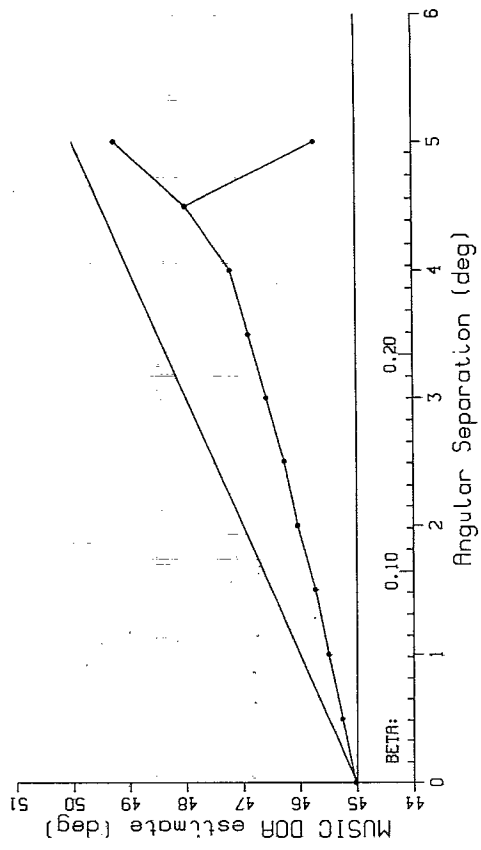
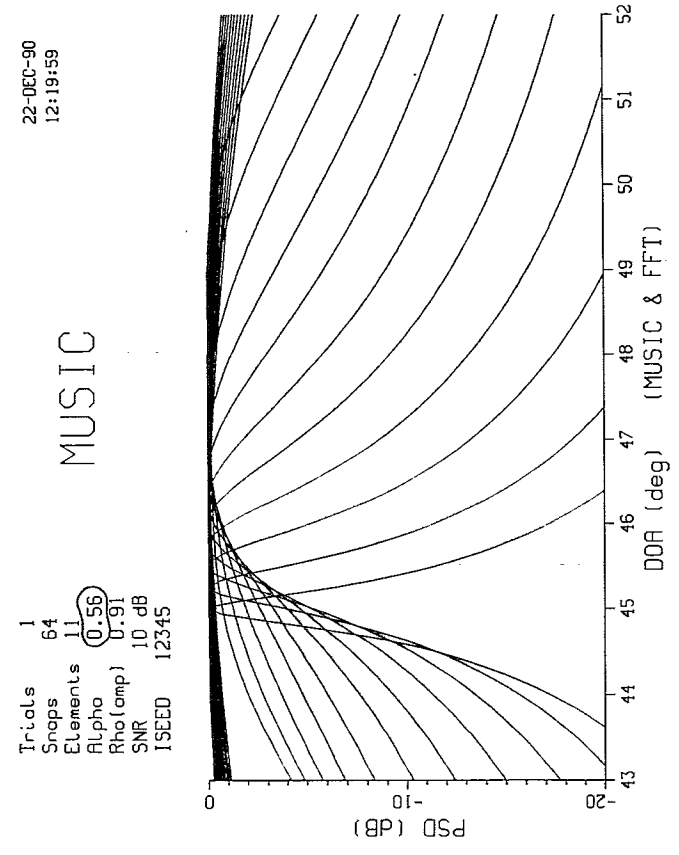


Fig. B1.4 - MUSIC algorithm; high correlation (α=0.57); 64 snapshots; 10 dB SNR



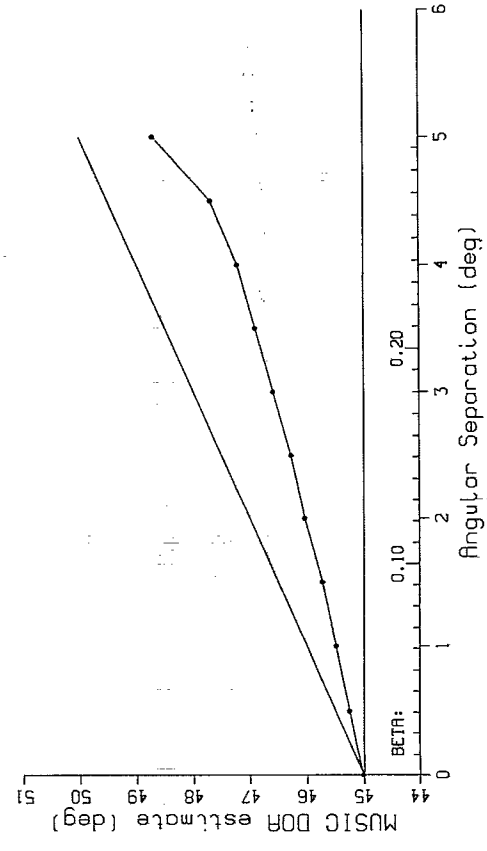


Fig. B1.3 - MUSIC algorithm; high correlation (α=0.56); 64 snapshots; 10 dB SNR

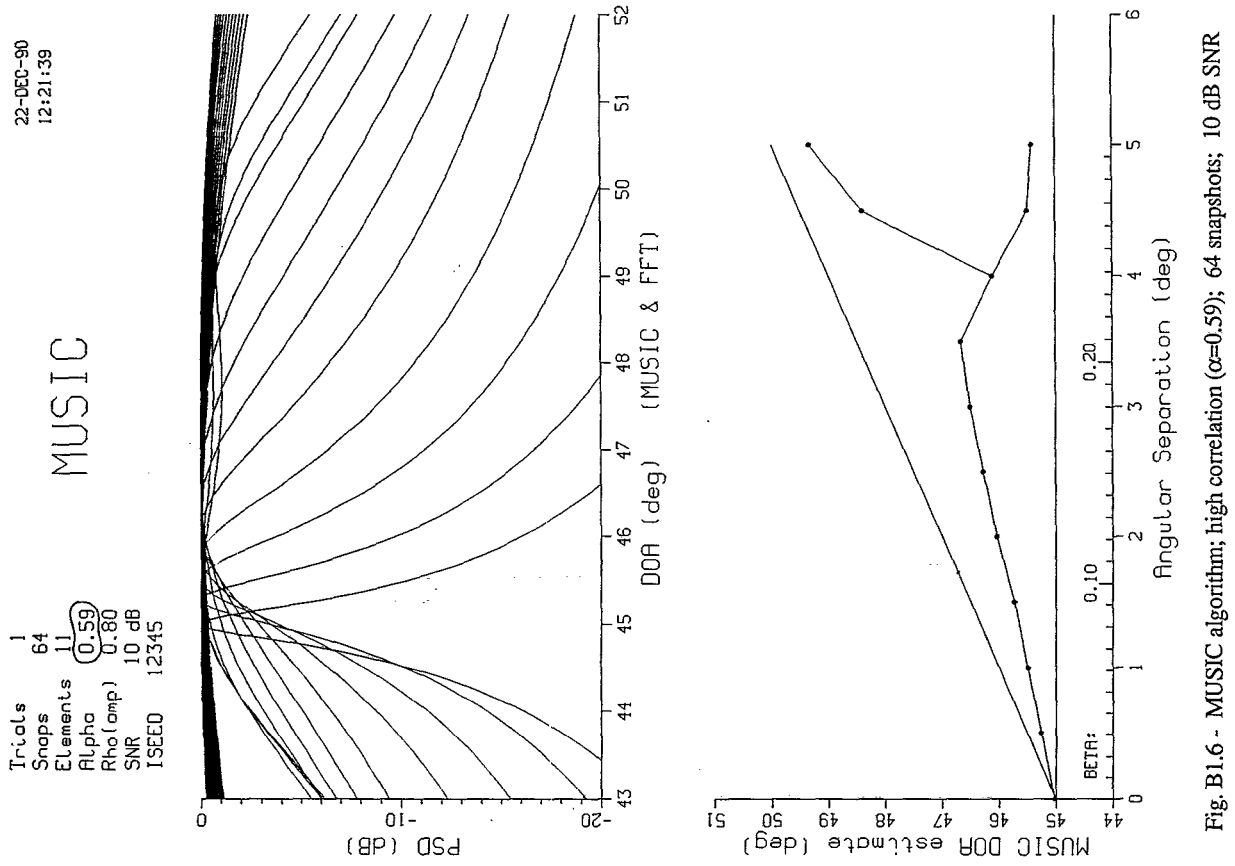


Fig. B1.5 - MUSIC algorithm; high correlation (α=0.58); 64 snapshots; 10 dB SNR

Fig. B1.6 - MUSIC algorithm; high correlation (α=0.59); 64 snapshots; 10 dB SNR

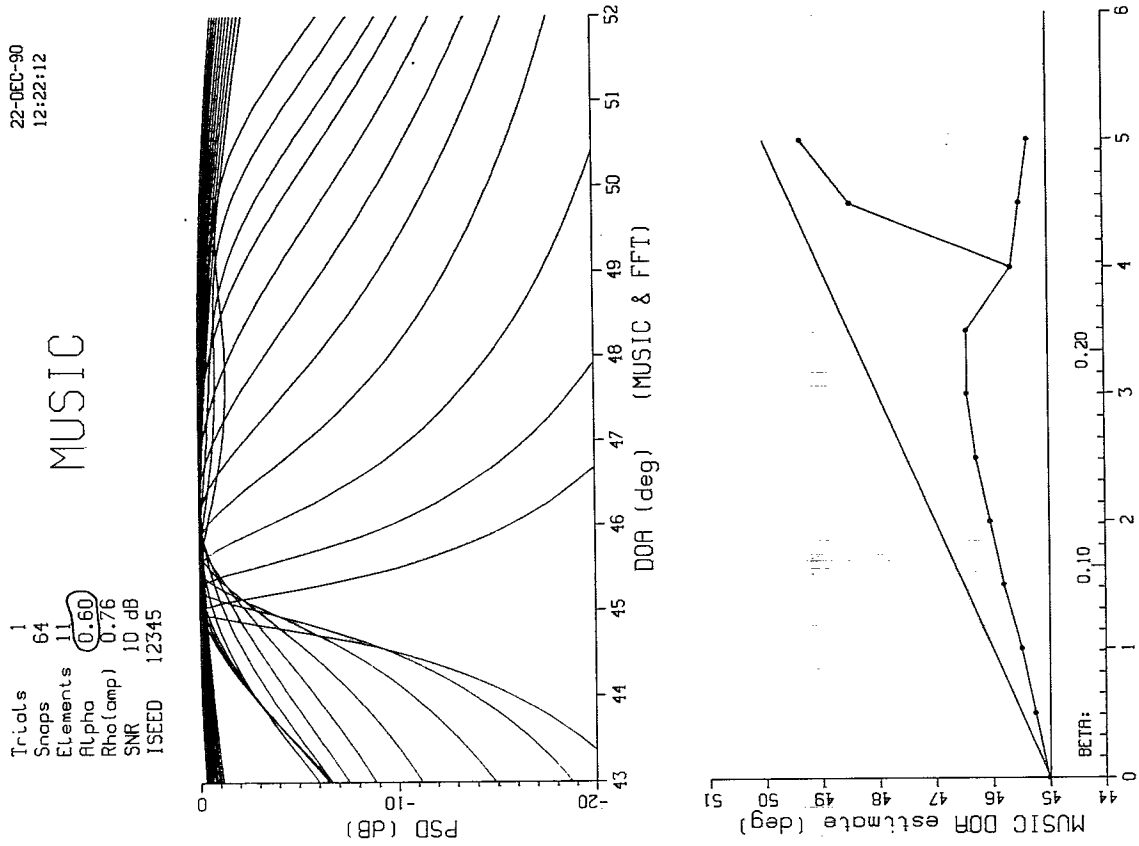


Fig. B1.7 - MUSIC algorithm; high correlation (α=0.60); 64 snapshots; 10 dB SNR

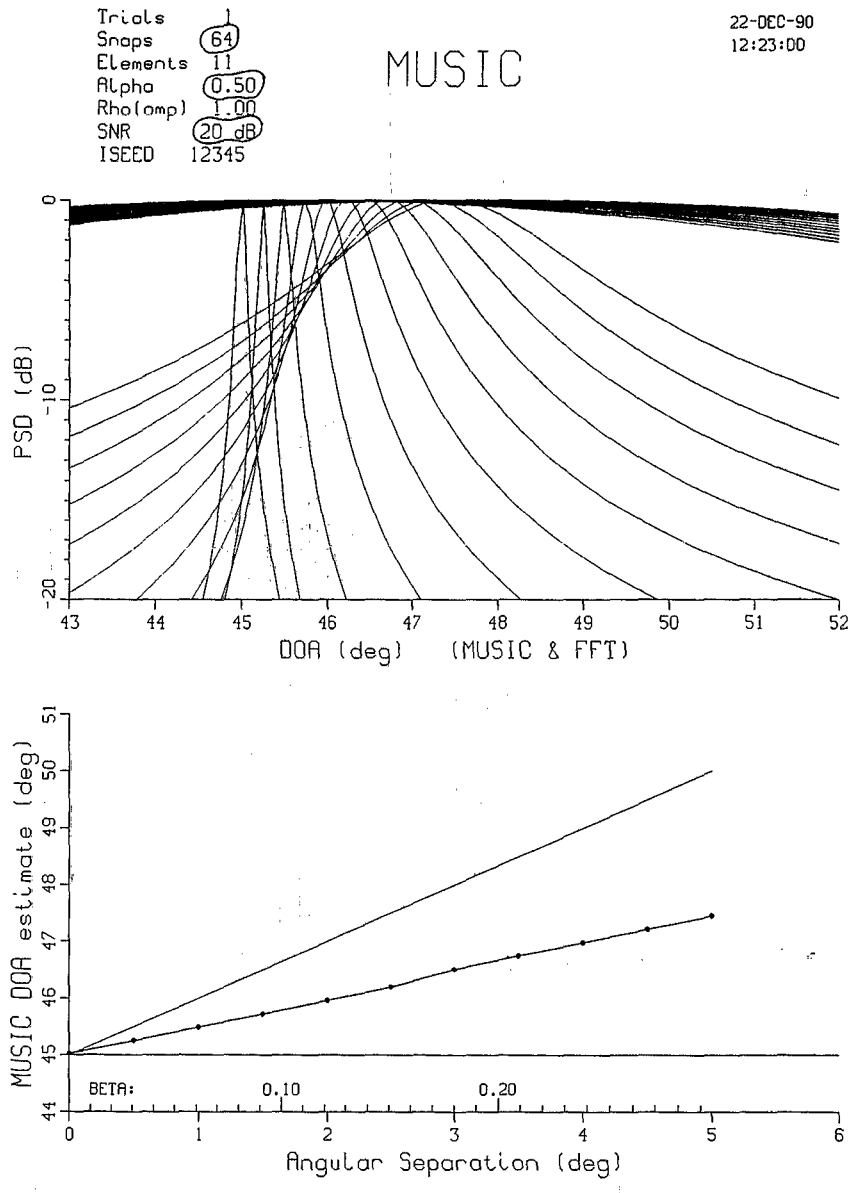


Fig. B2.1 - MUSIC algorithm; high correlation (α=0.50); 64 snapshots; 20 dB SNR

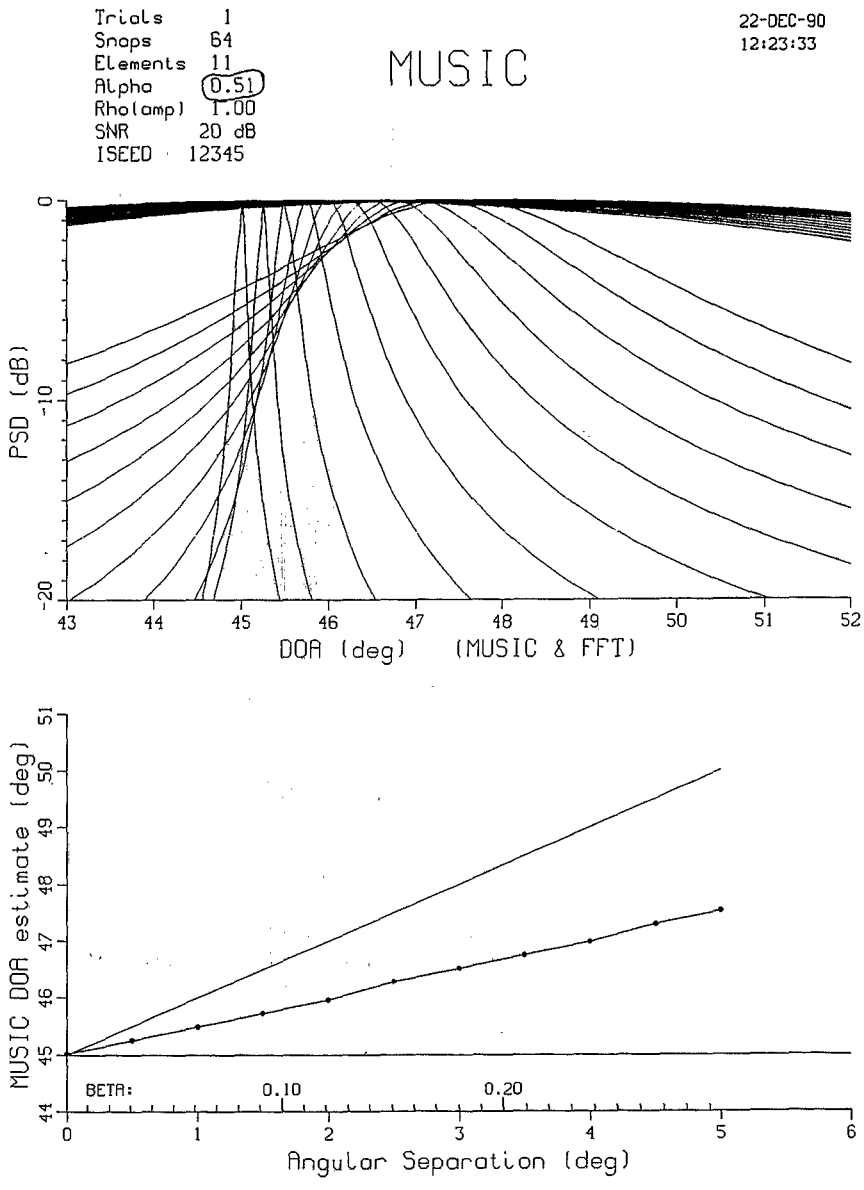
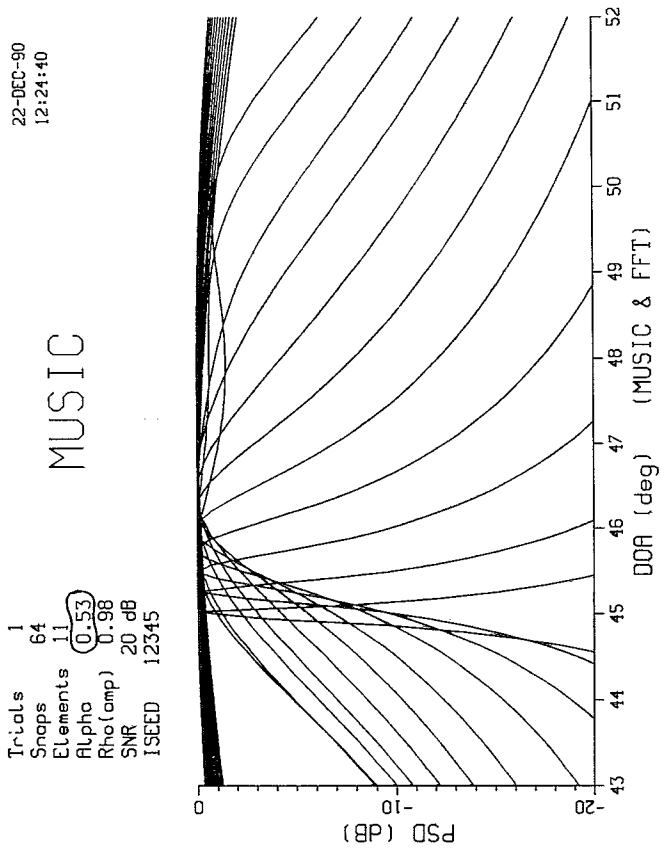


Fig. B2.2 - MUSIC algorithm; high correlation (α=0.51); 64 snapshots; 20 dB SNR



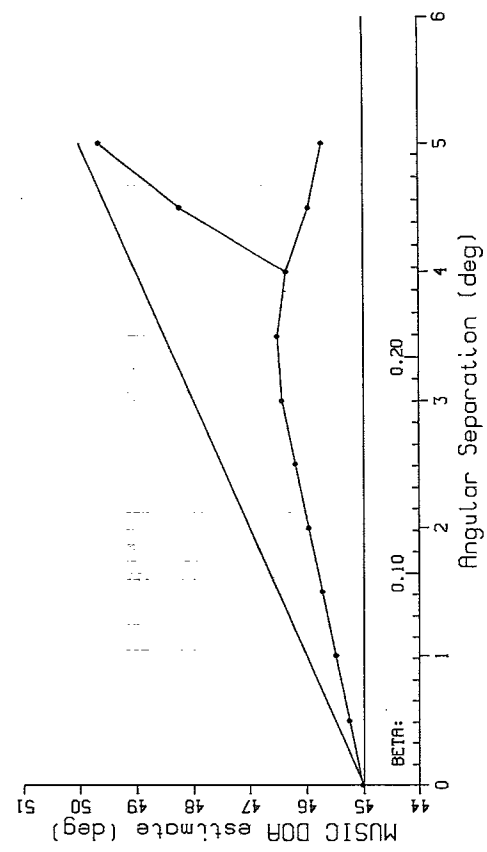
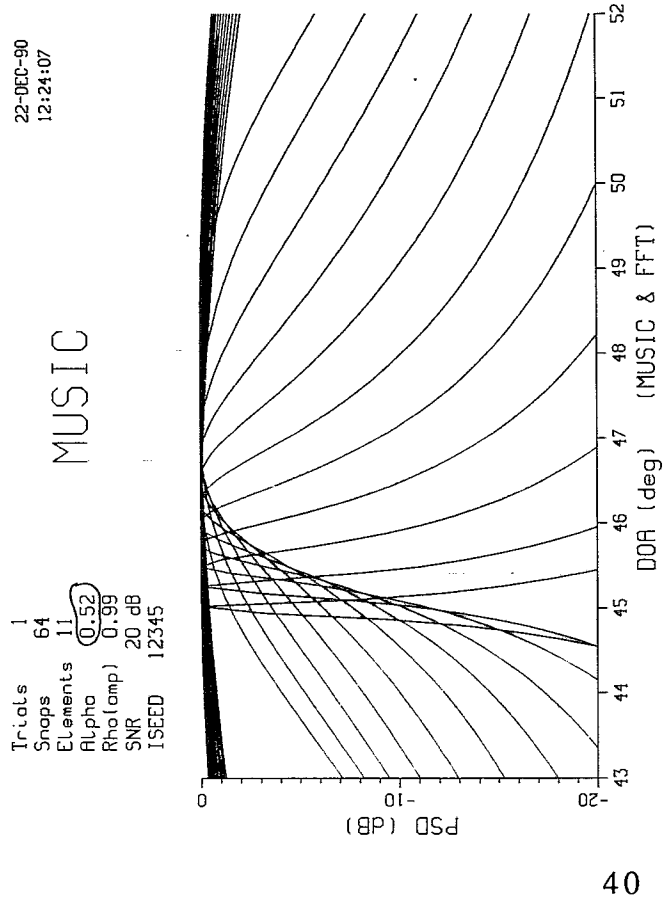


Fig. B2.3 - MUSIC algorithm; high correlation (α=0.52); 64 snapshots; 20 dB SNR



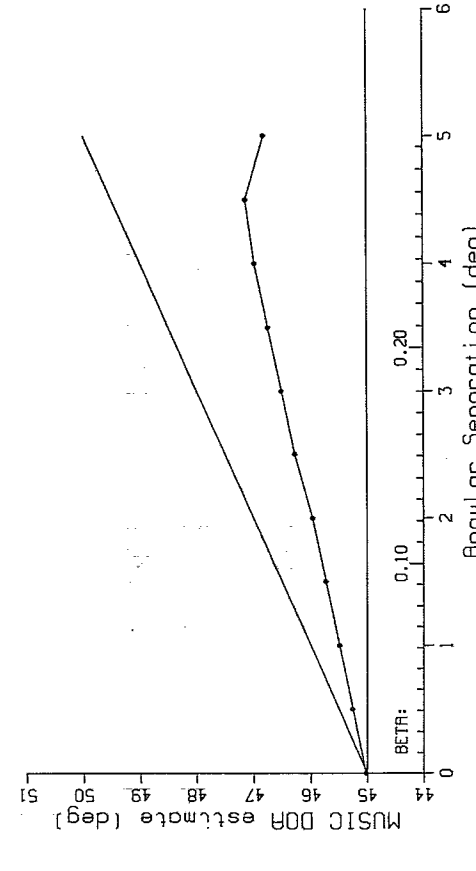


Fig. B2.4 - MUSIC algorithm; high correlation (α=0.53); 64 snapshots; 20 dB SNR

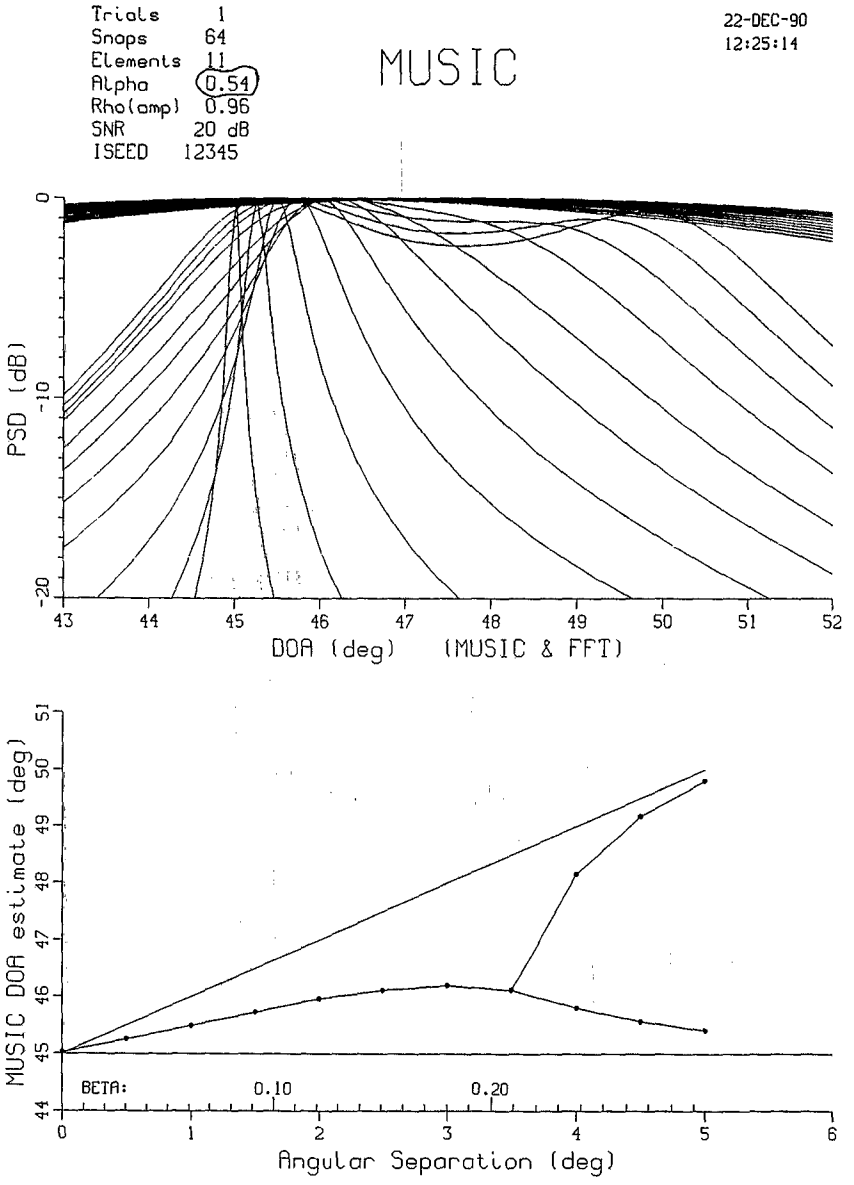


Fig. B2.5 - MUSIC algorithm; high correlation (α=0.54); 64 snapshots; 20 dB SNR

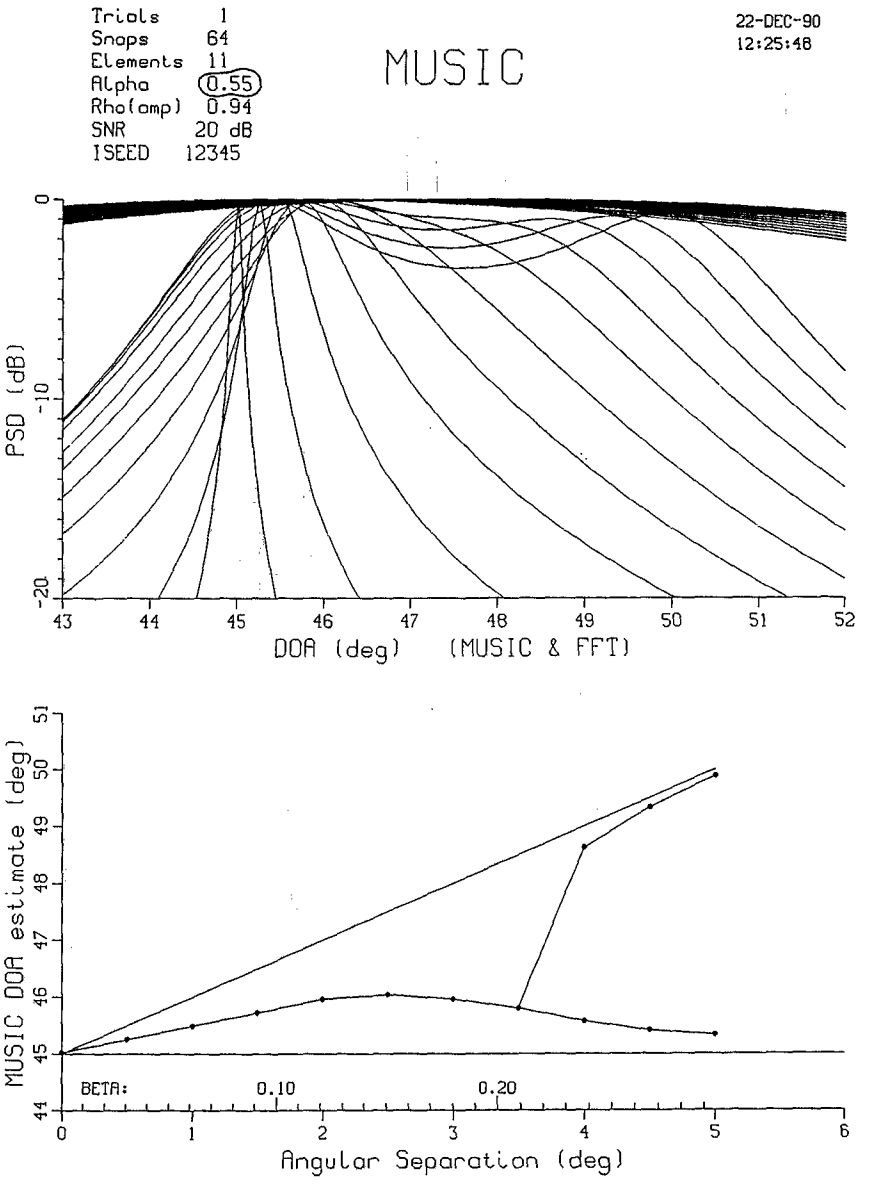


Fig. B2.6 - MUSIC algorithm; high correlation (α=0.55); 64 snapshots; 20 dB SNR

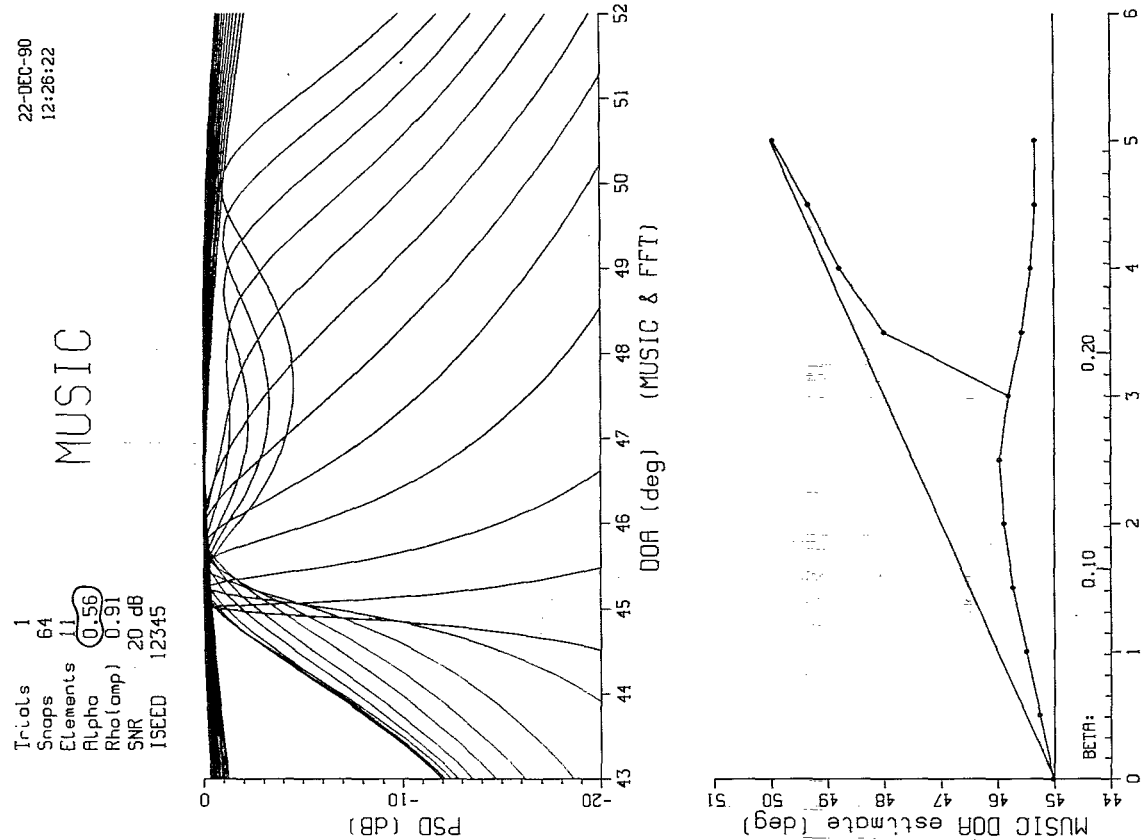


Fig. B2.7 - MUSIC algorithm; high correlation (α=0.56); 64 snapshots; 20 dB SNR

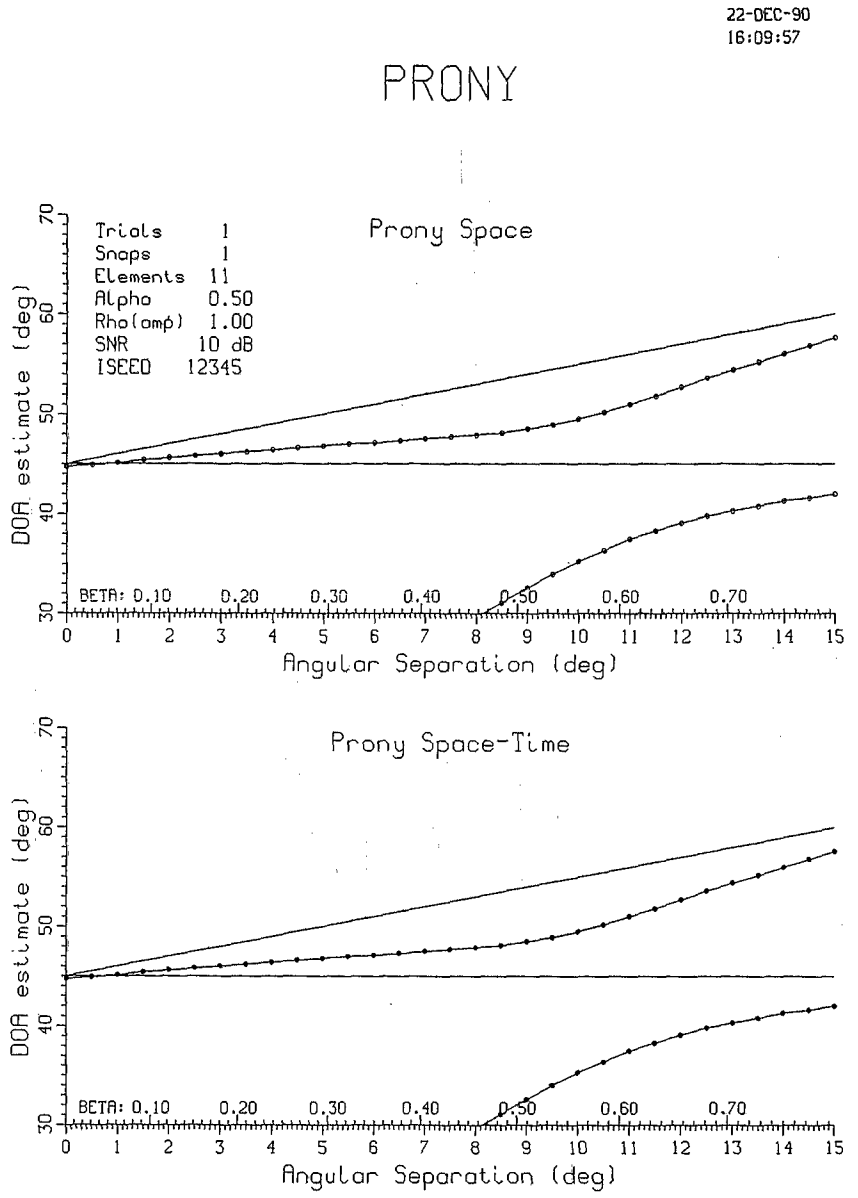


Fig. C1.1 - PRONY algorithm; perfect correlation; 10 dB SNR; 1 snapshot

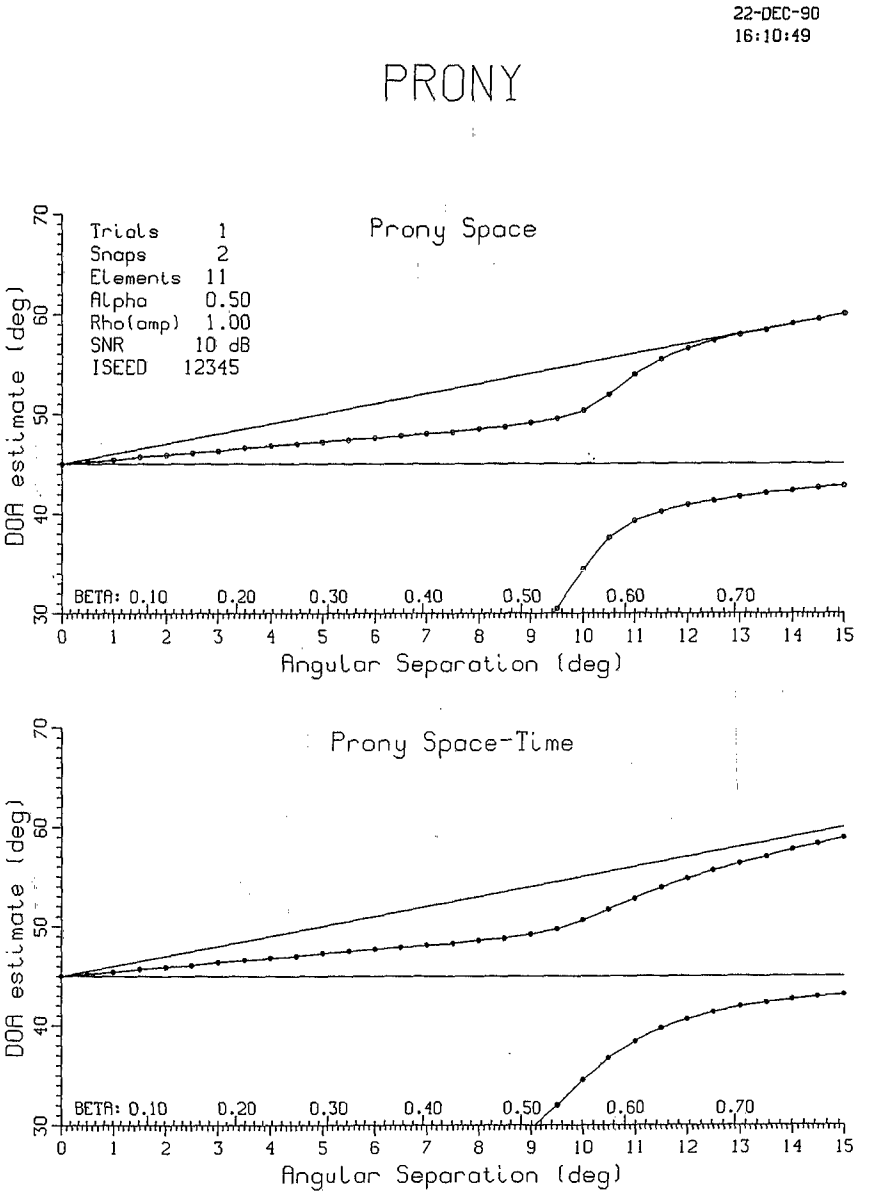


Fig. C1.2 - PRONY algorithm; perfect correlation; 10 dB SNR; 2 snapshots

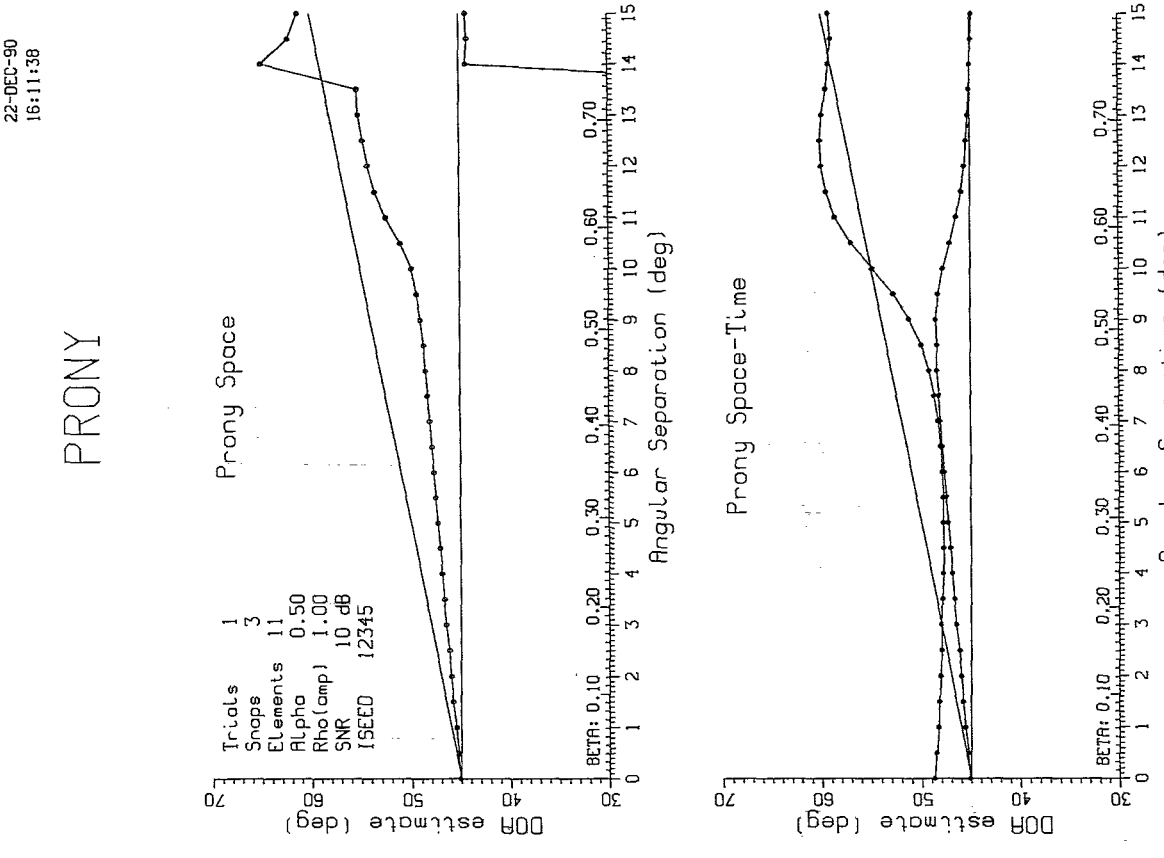


Fig. C1.3 - PRONY algorithm; perfect correlation; 10 dB SNR; 3 snapshots

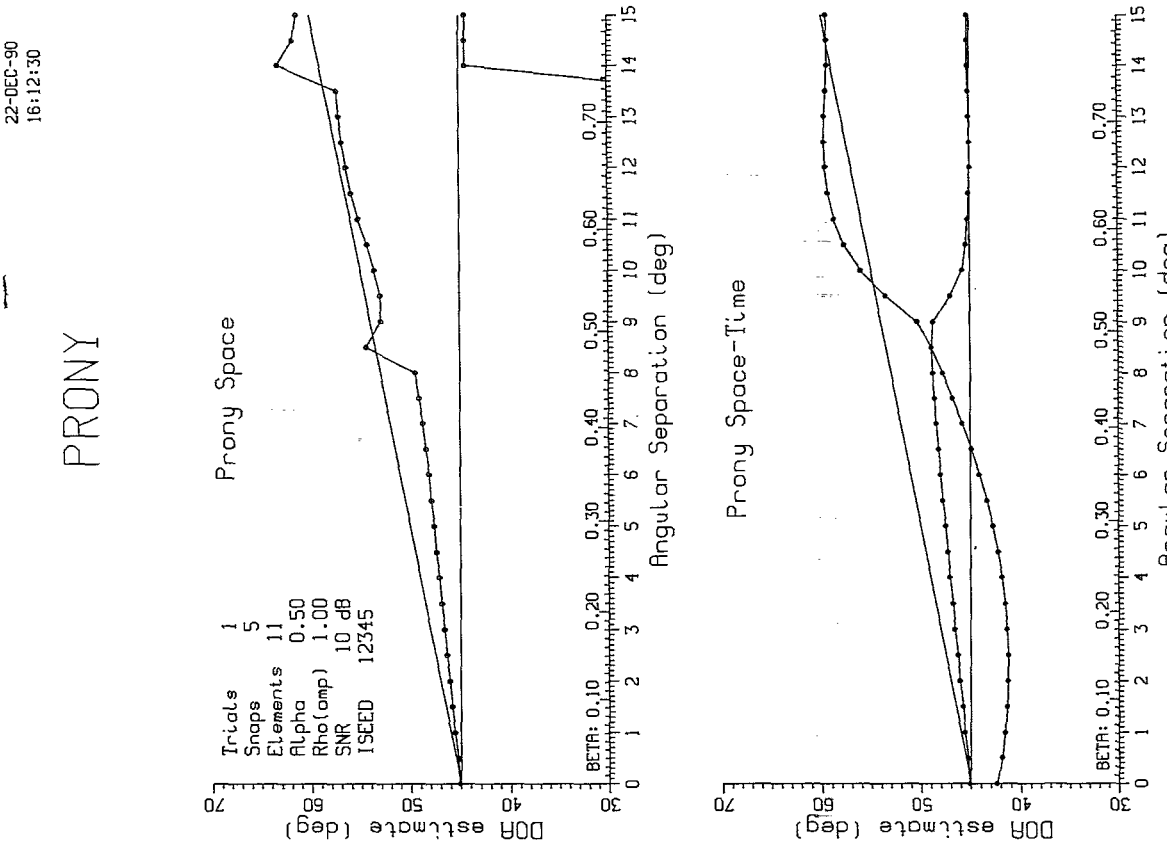


Fig. C1.4 - PRONY algorithm; perfect correlation; 10 dB SNR; 5 snapshots

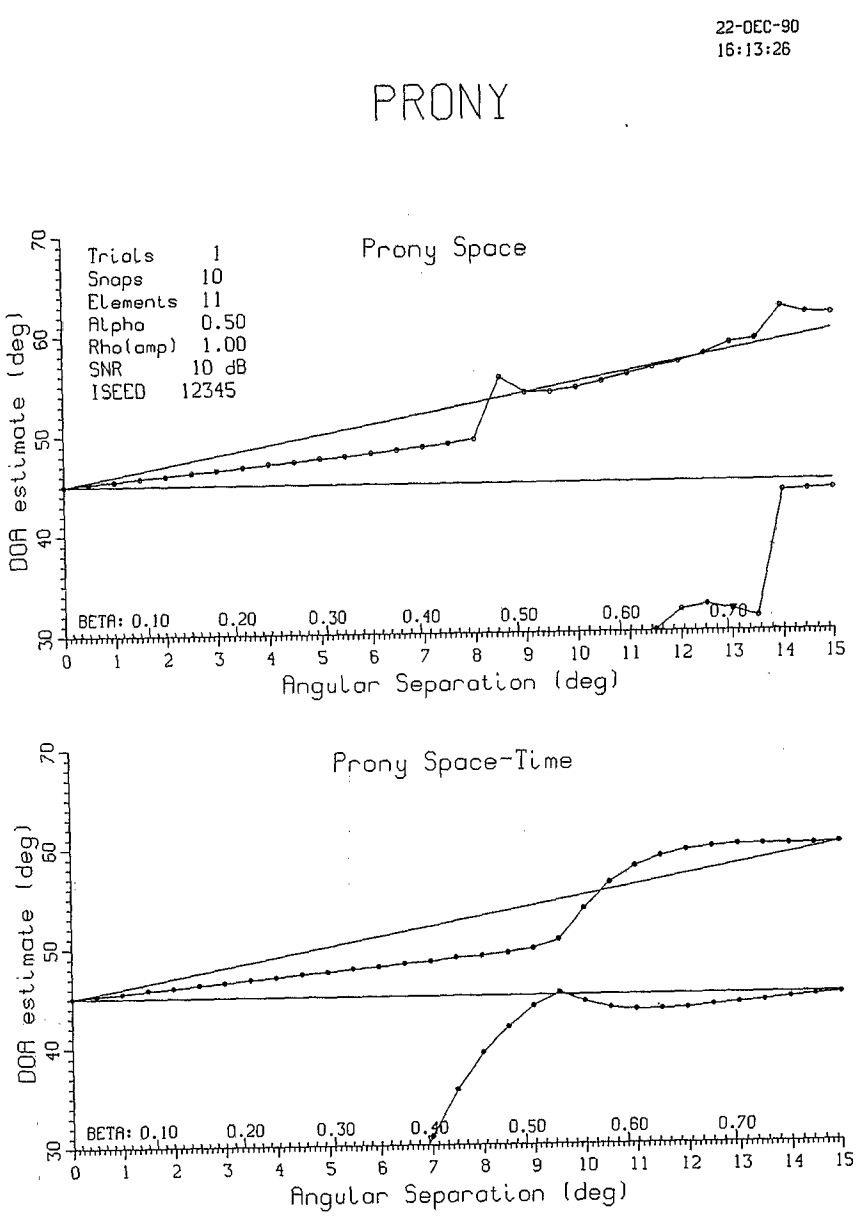


Fig. C1.5 - PRONY algorithm; perfect correlation; 10 dB SNR; 10 snapshots

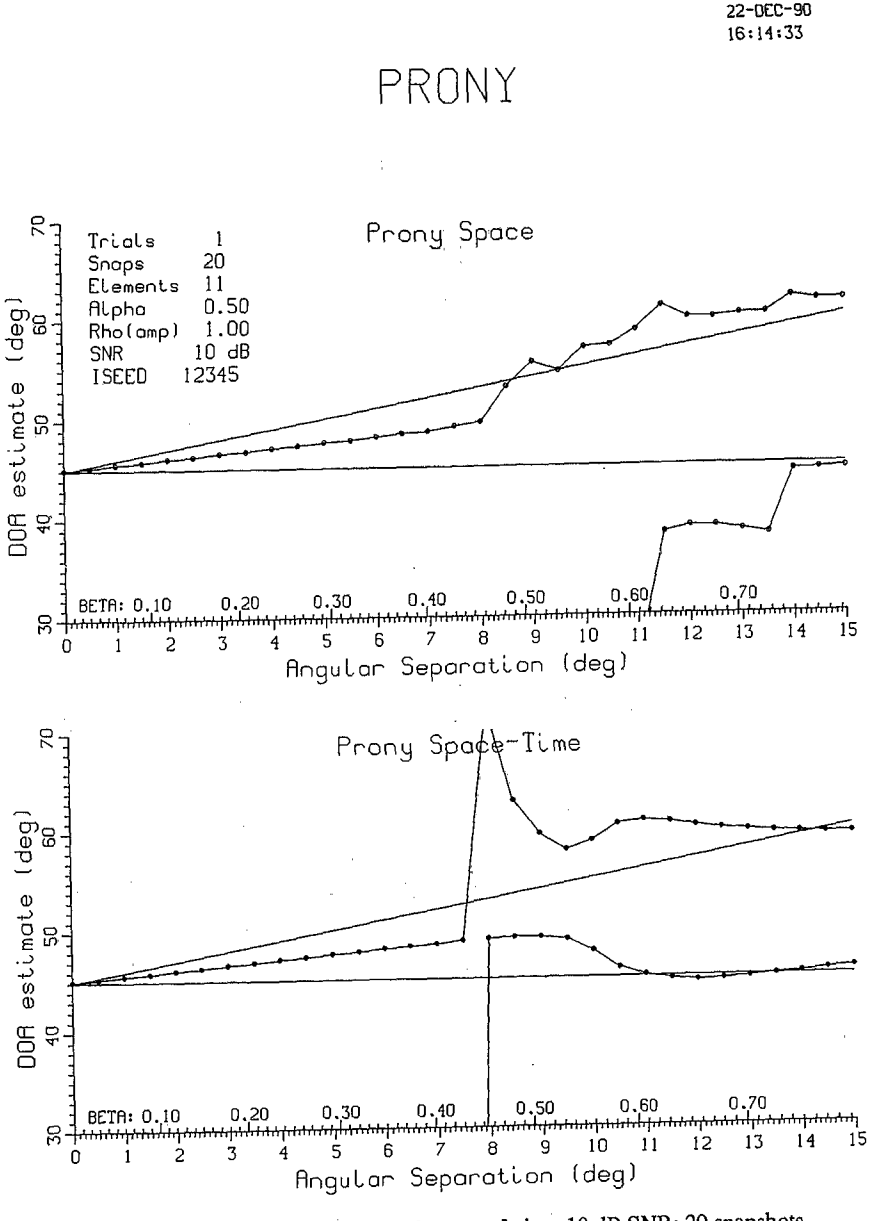


Fig. C1.6 - PRONY algorithm; perfect correlation; 10 dB SNR; 20 snapshots

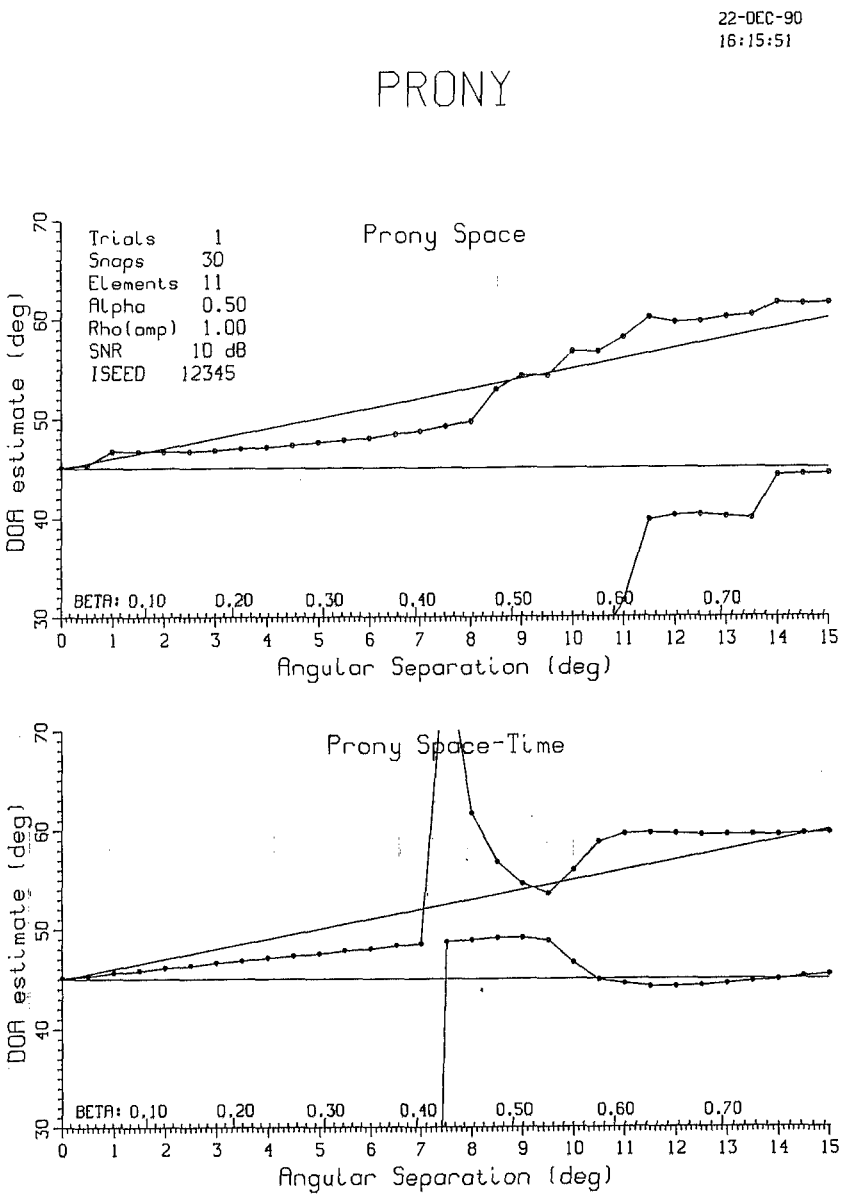


Fig. C1.7 - PRONY algorithm; perfect correlation; 10 dB SNR; 30 snapshots

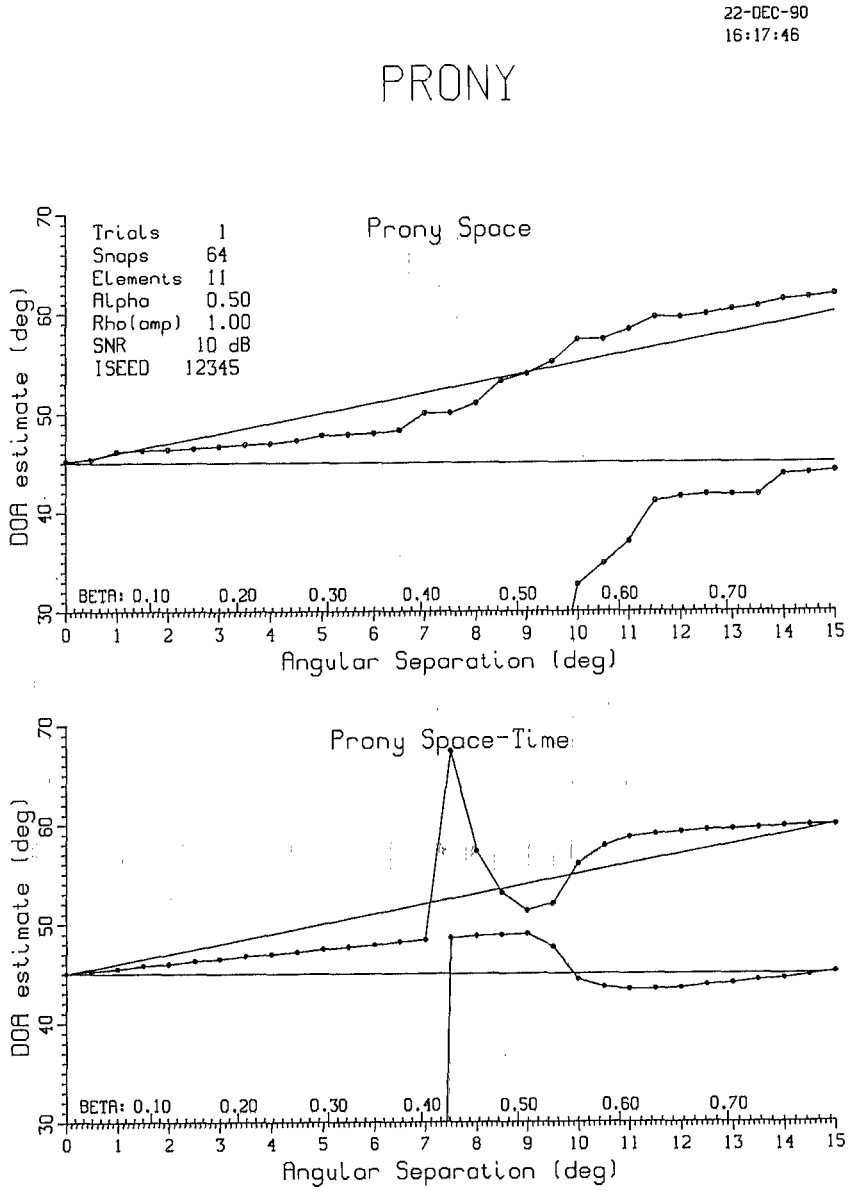


Fig. C1.8 - PRONY algorithm; perfect correlation; 10 dB SNR; 64 snapshots

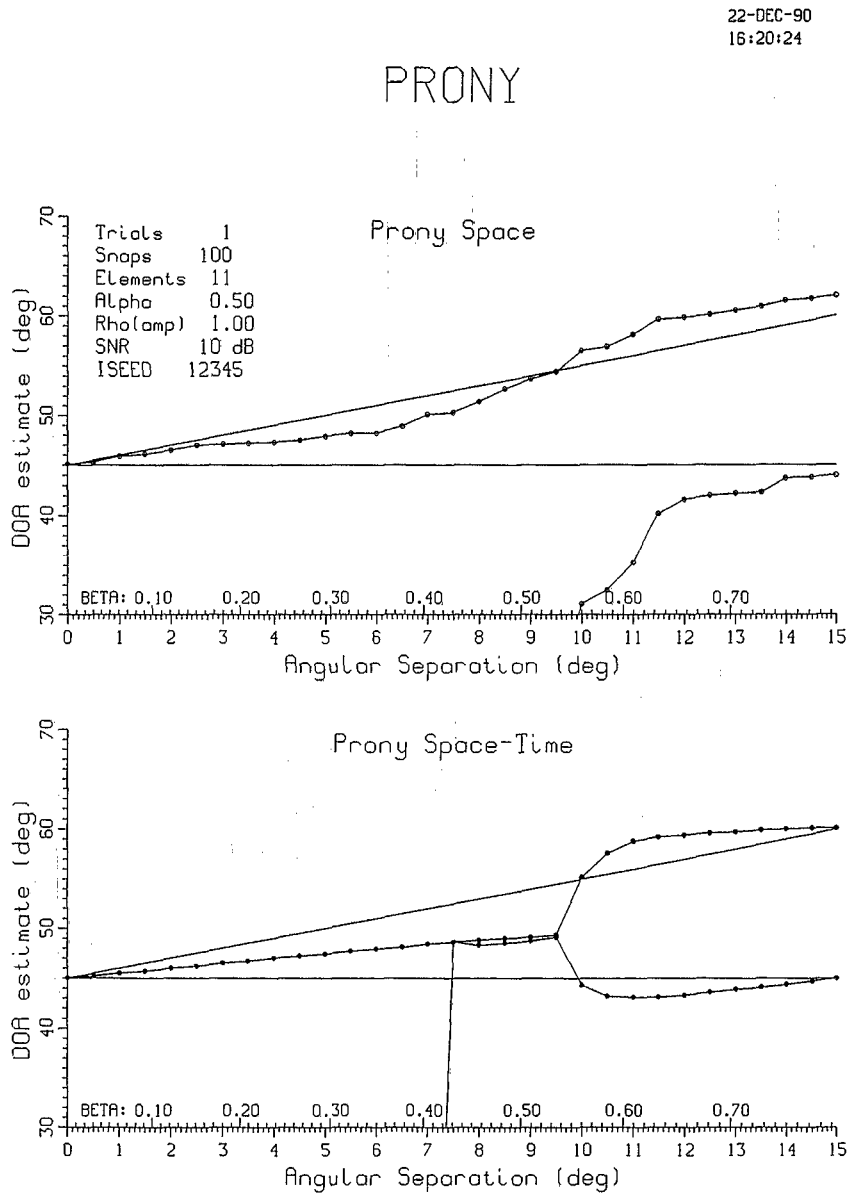


Fig. C1.9 - PRONY algorithm; perfect correlation; 10 dB SNR; 100 snapshots

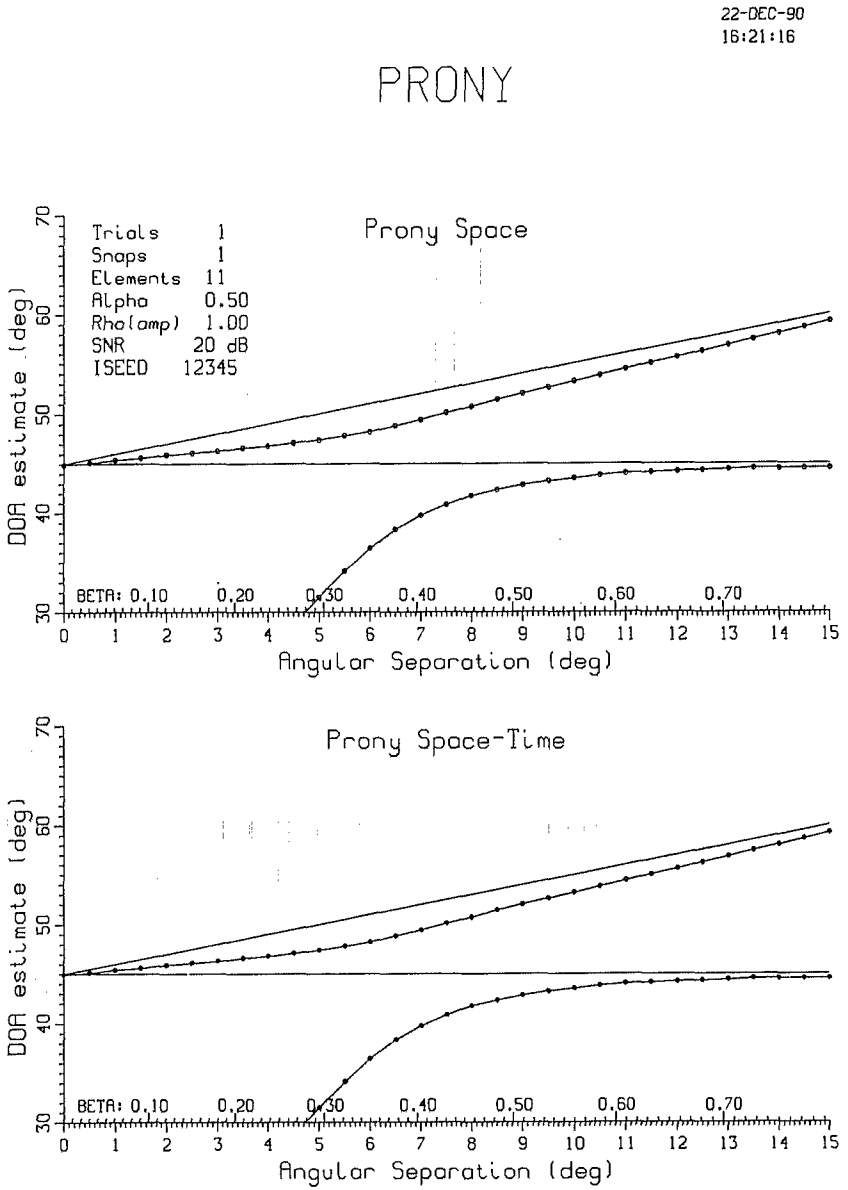


Fig. C2.1 - PRONY algorithm; perfect correlation; 20 dB SNR; 1 snapshot

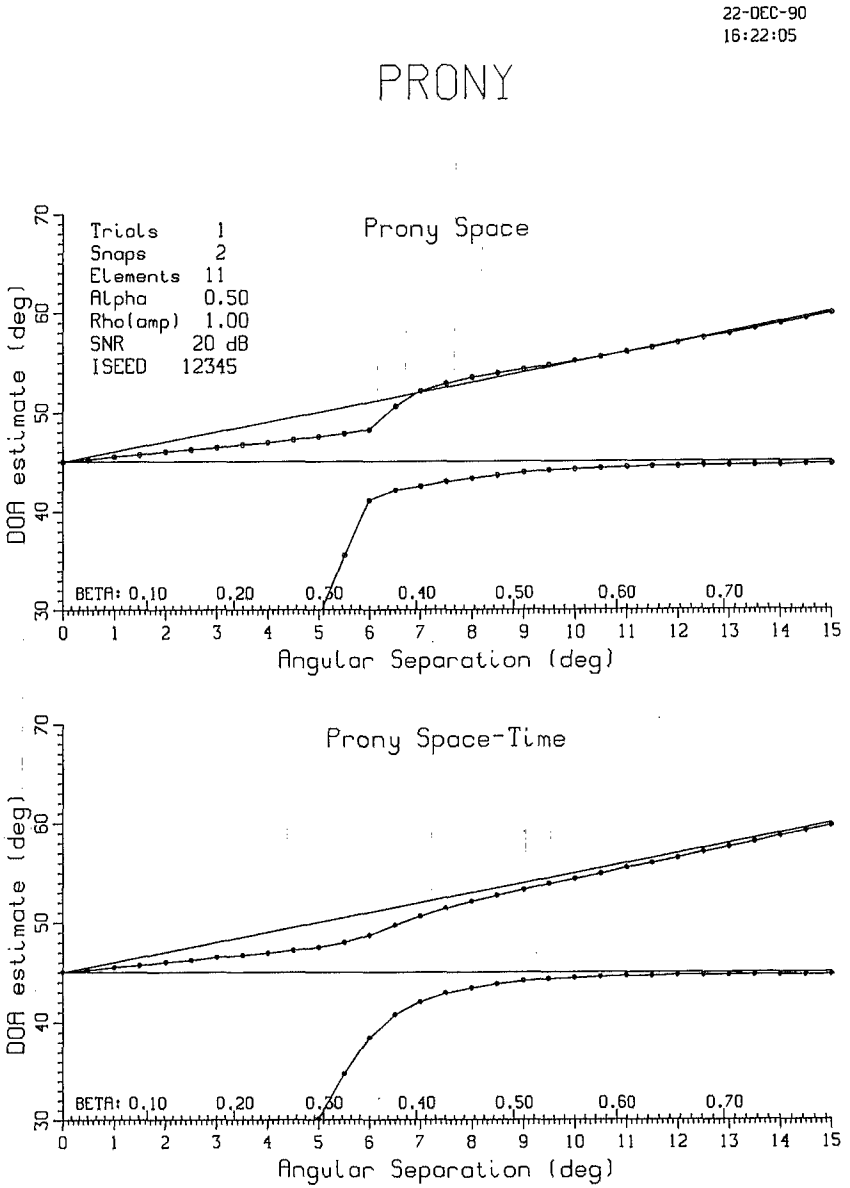


Fig. C2.2 - PRONY algorithm; perfect correlation; 20 dB SNR; 2 snapshots

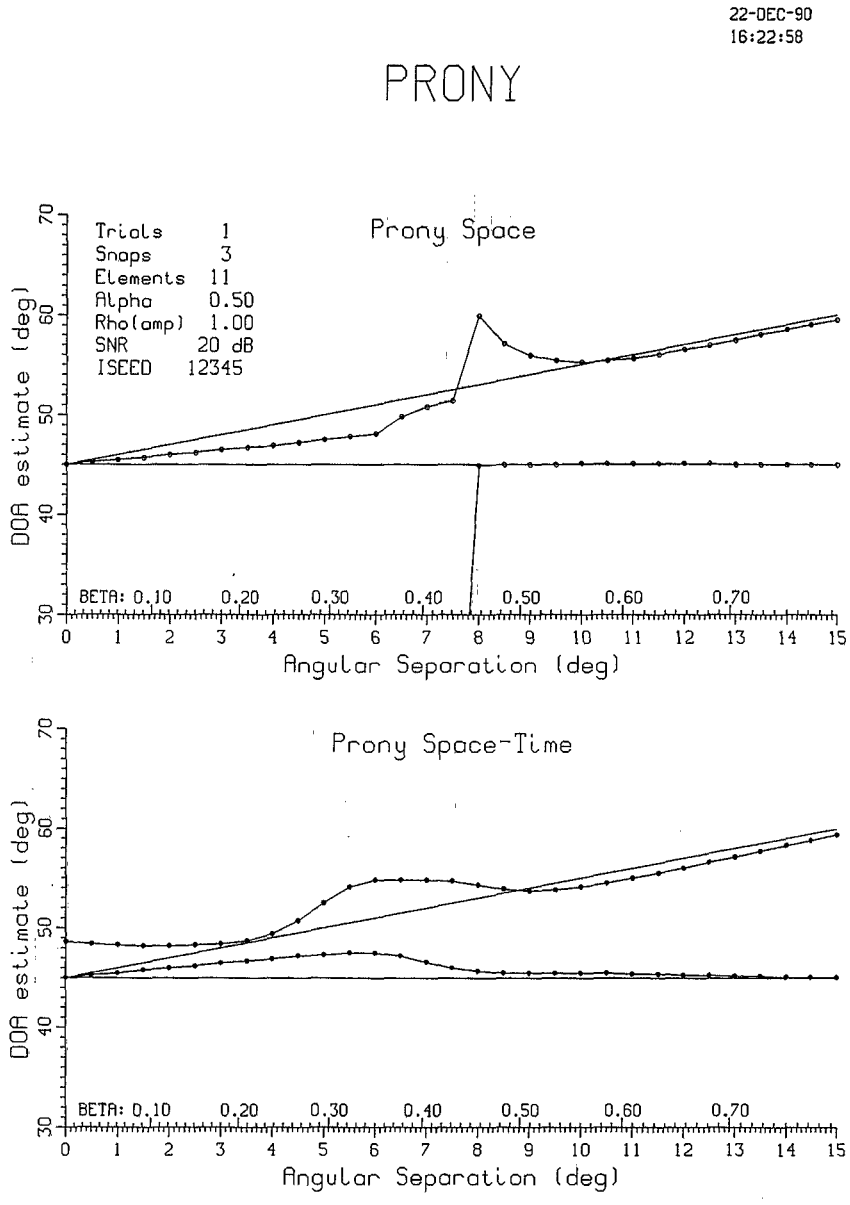


Fig. C2.3 - PRONY algorithm; perfect correlation; 20 dB SNR; 3 snapshots

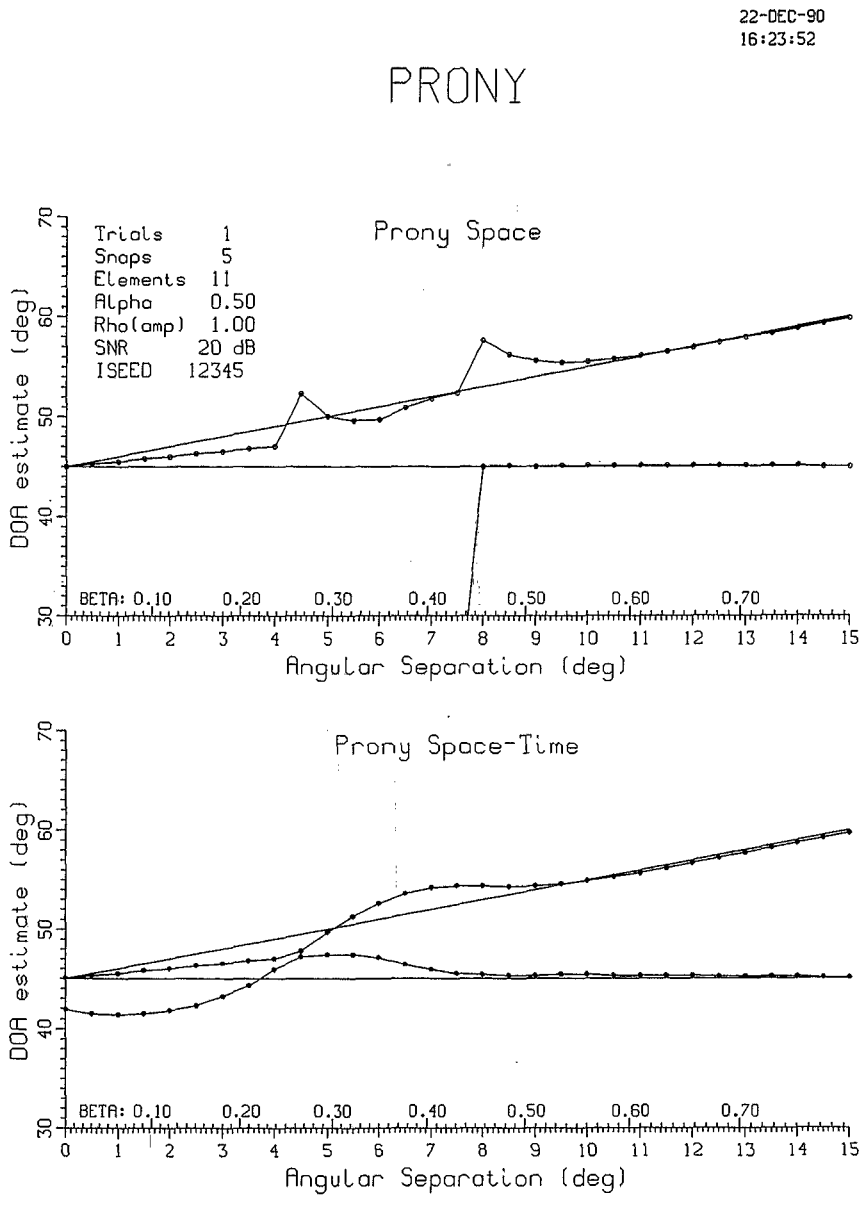
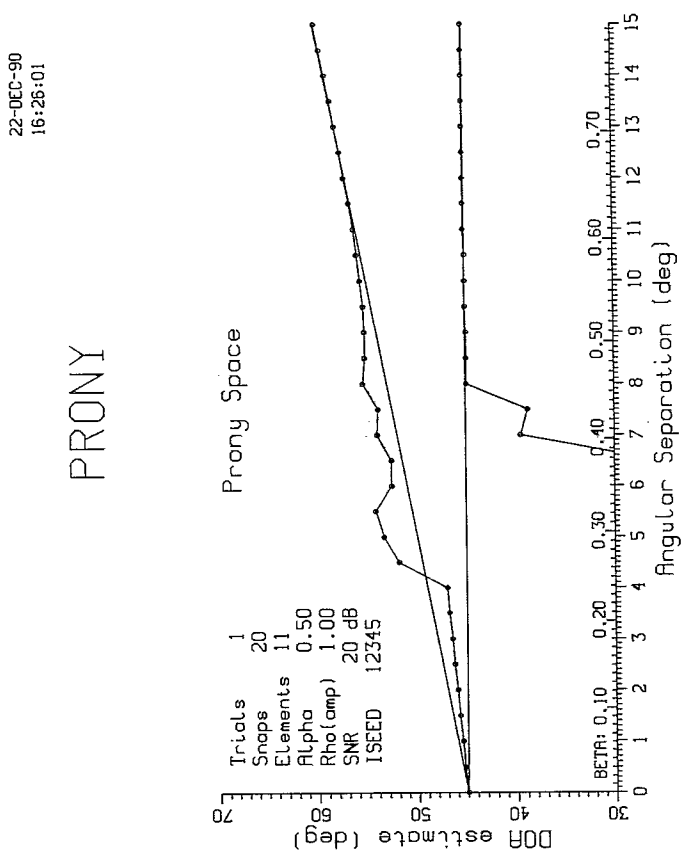


Fig. C2.4 - PRONY algorithm; perfect correlation; 20 dB SNR; 5 snapshots



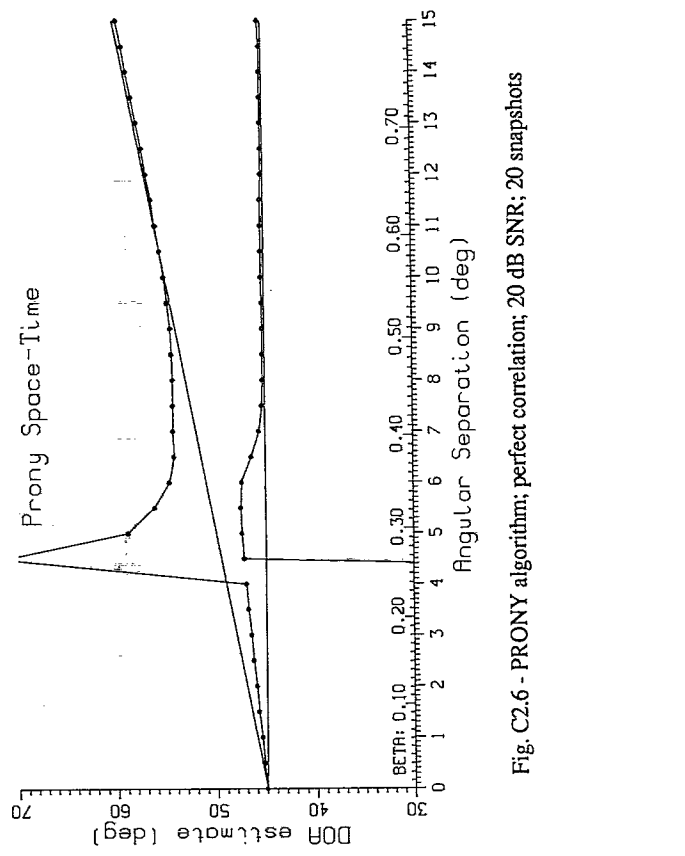
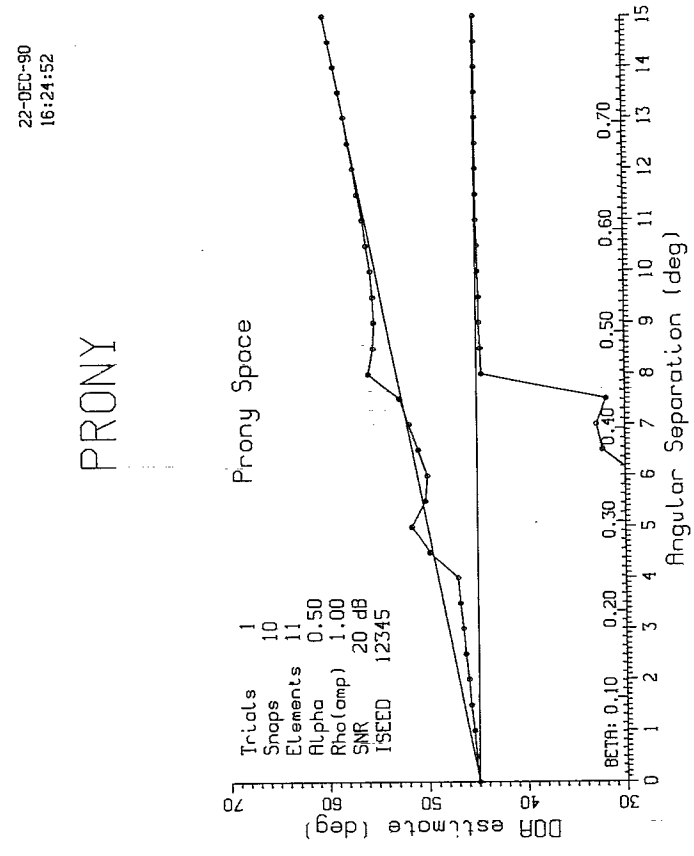


Fig. C2.6 - PRONY algorithm; perfect correlation; 20 dB SNR; 20 snapshots



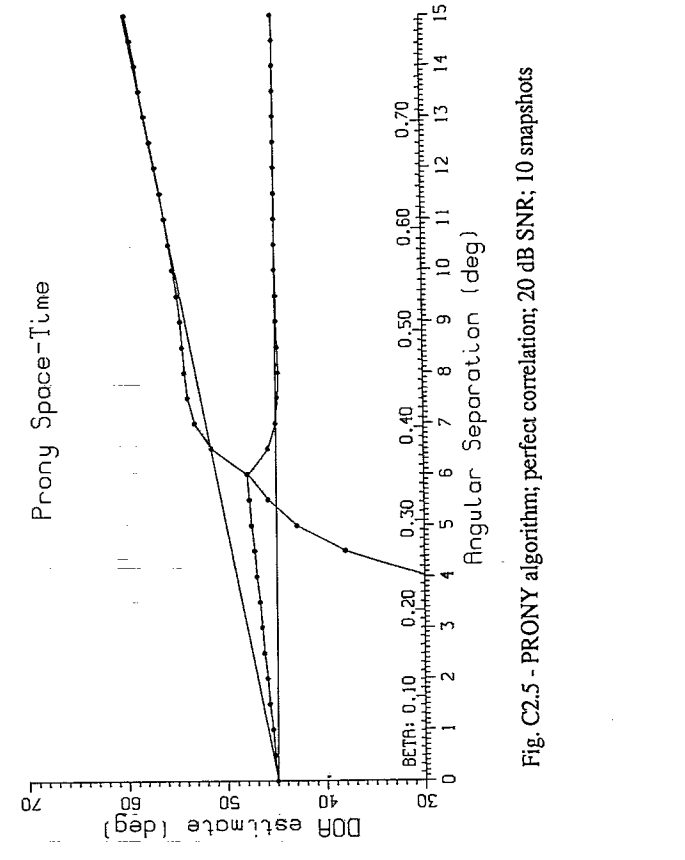


Fig. C2.5 - PRONY algorithm; perfect correlation; 20 dB SNR; 10 snapshots

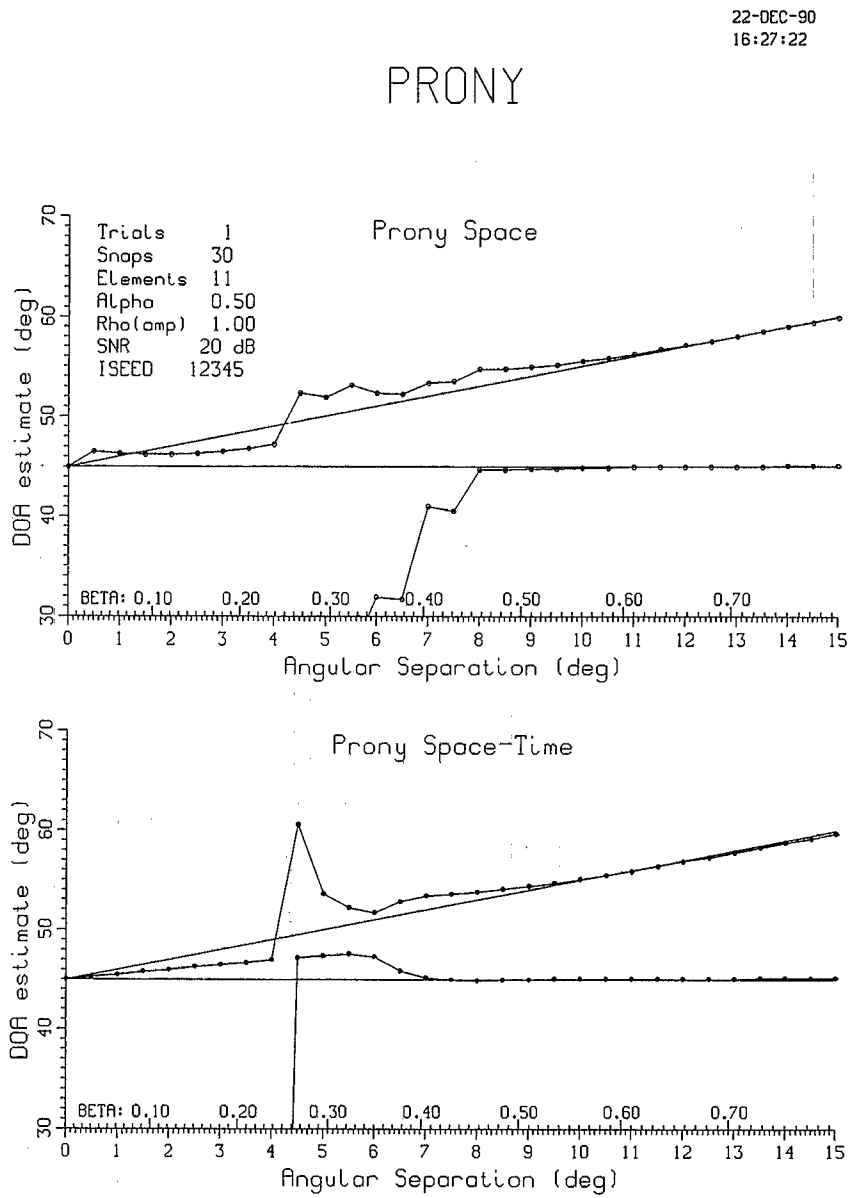


Fig. C2.7 - PRONY algorithm; perfect correlation; 20 dB SNR; 30 snapshots

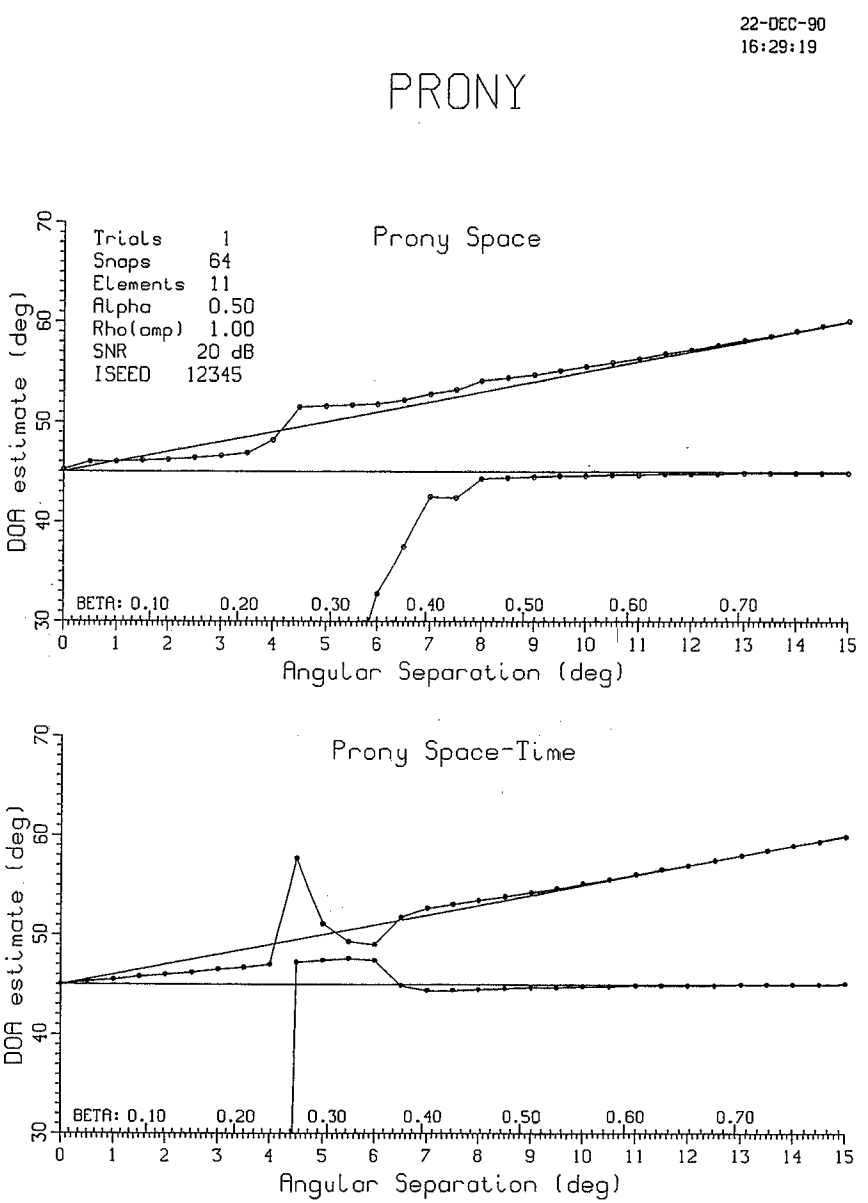


Fig. C2.8 - PRONY algorithm; perfect correlation; 20 dB SNR; 64 snapshots

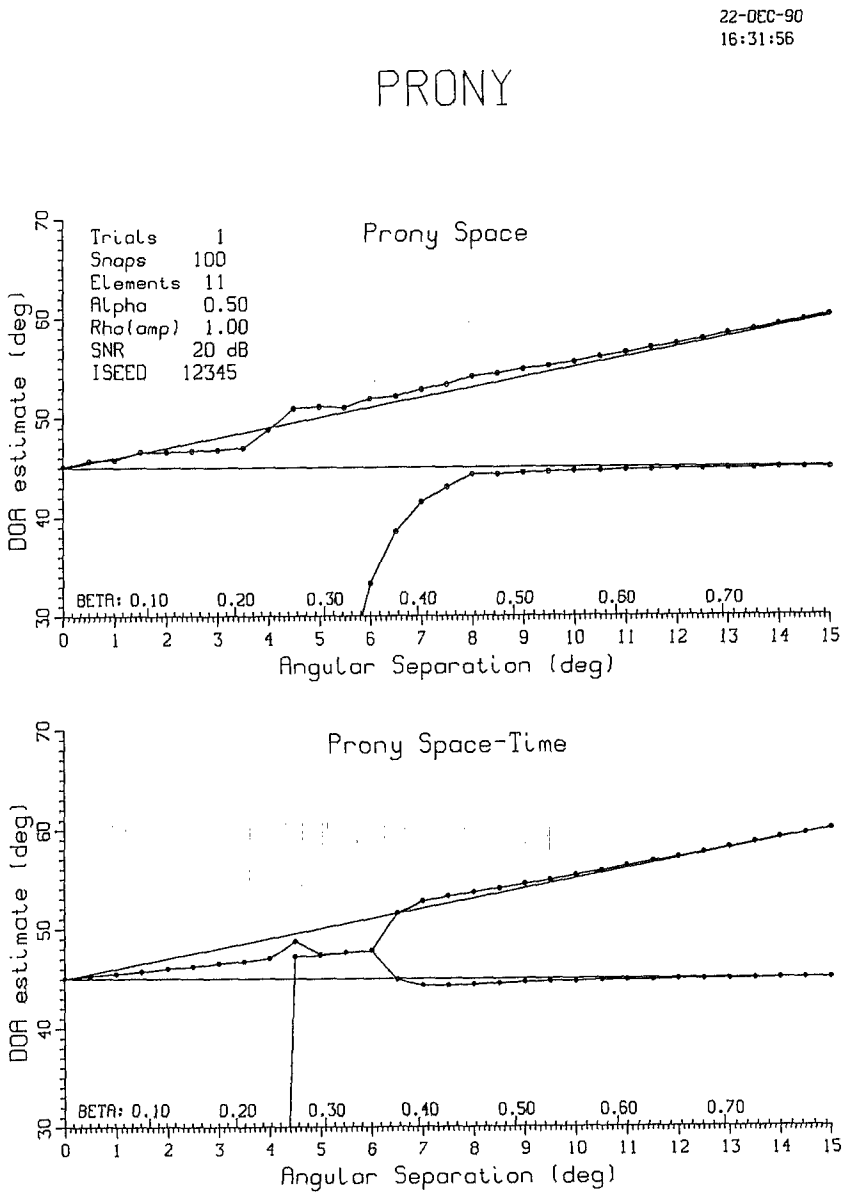


Fig. C2.9 - PRONY algorithm; perfect correlation; 20 dB SNR; 100 snapshots

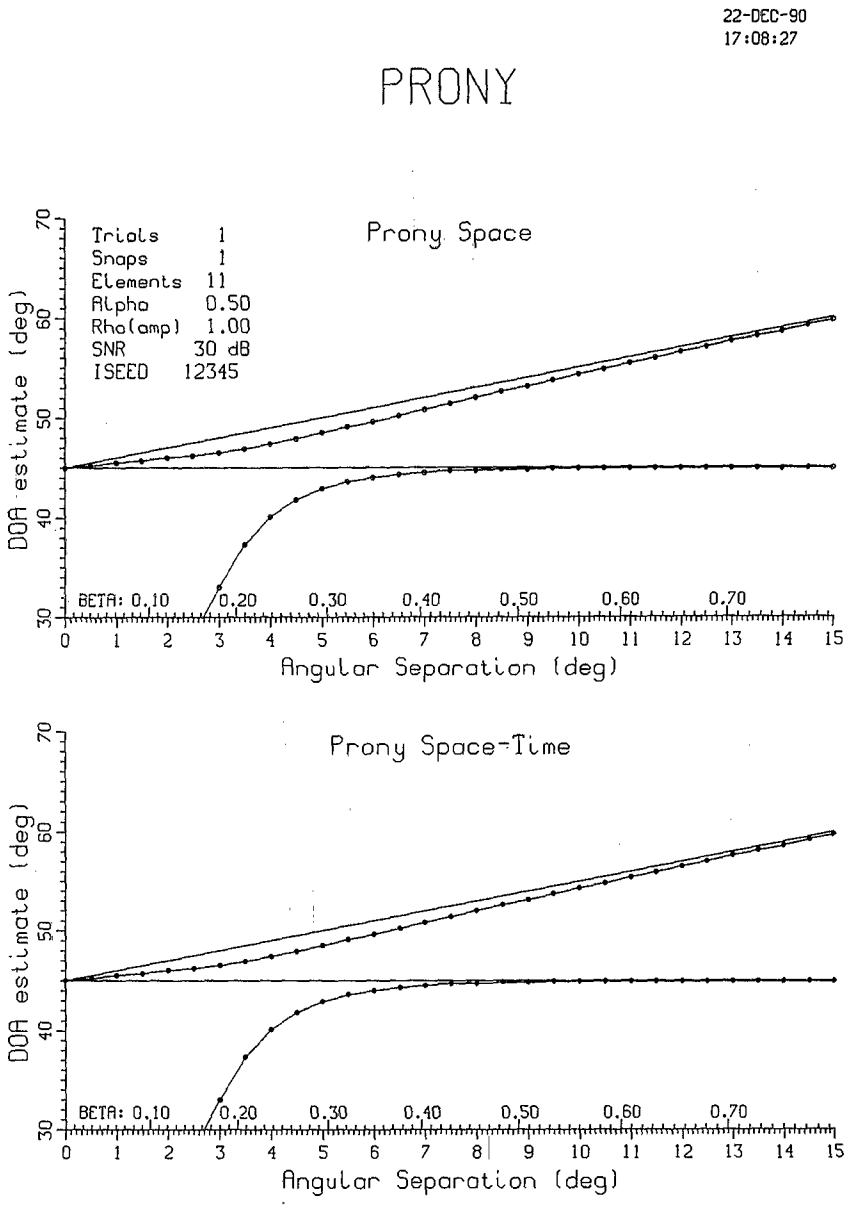


Fig. C3.1 - PRONY algorithm; perfect correlation; 30 dB SNR; 1 snapshot

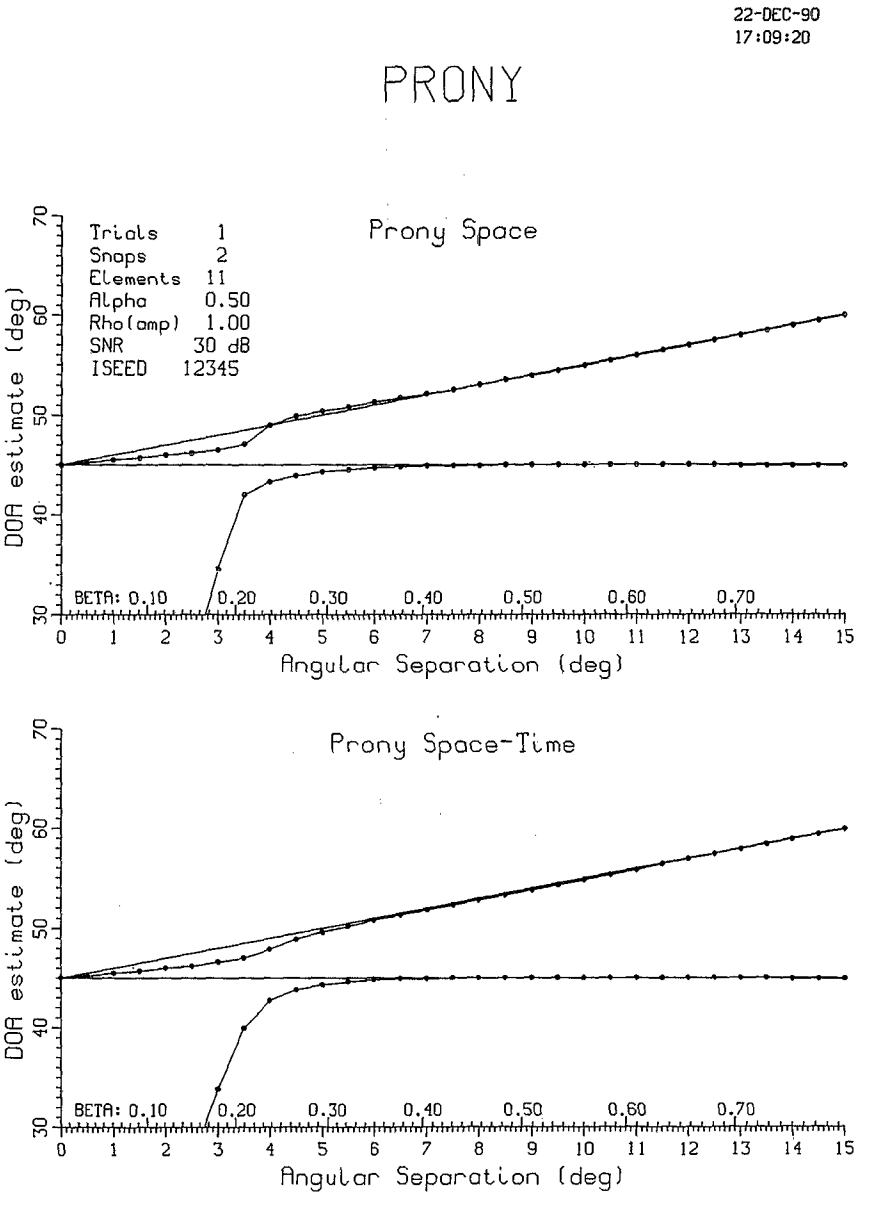


Fig. C3.2 - PRONY algorithm; perfect correlation; 30 dB SNR; 2 snapshots

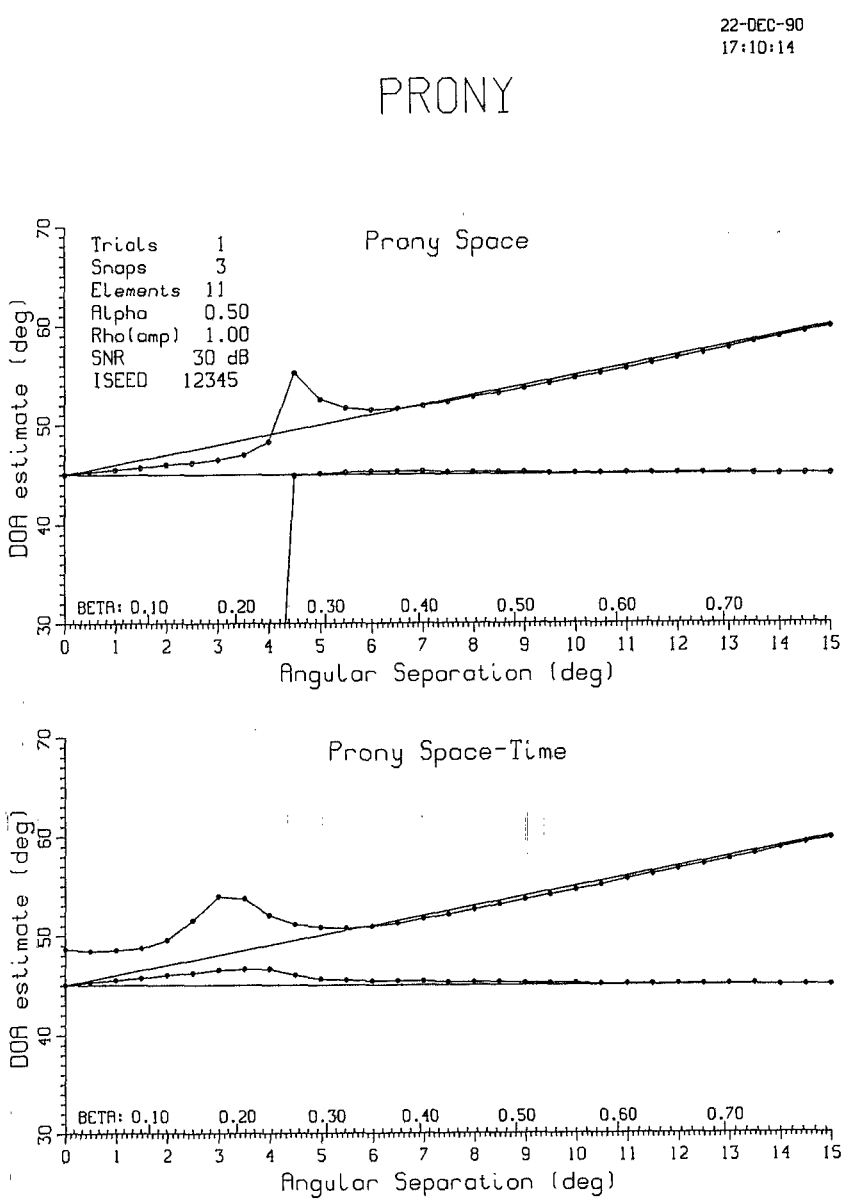


Fig. C3.3 - PRONY algorithm; perfect correlation; 30 dB SNR; 3 snapshots

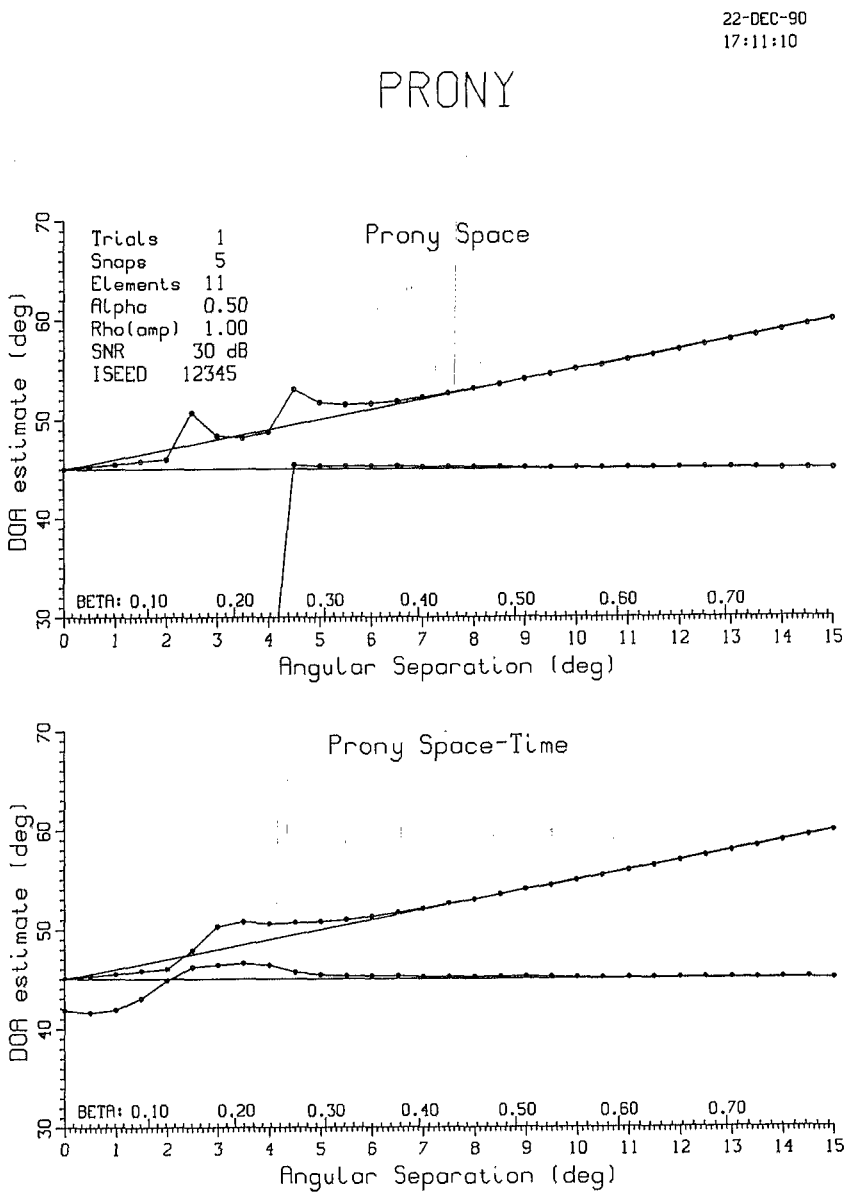


Fig. C3.4 - PRONY algorithm; perfect correlation; 30 dB SNR; 5 snapshots

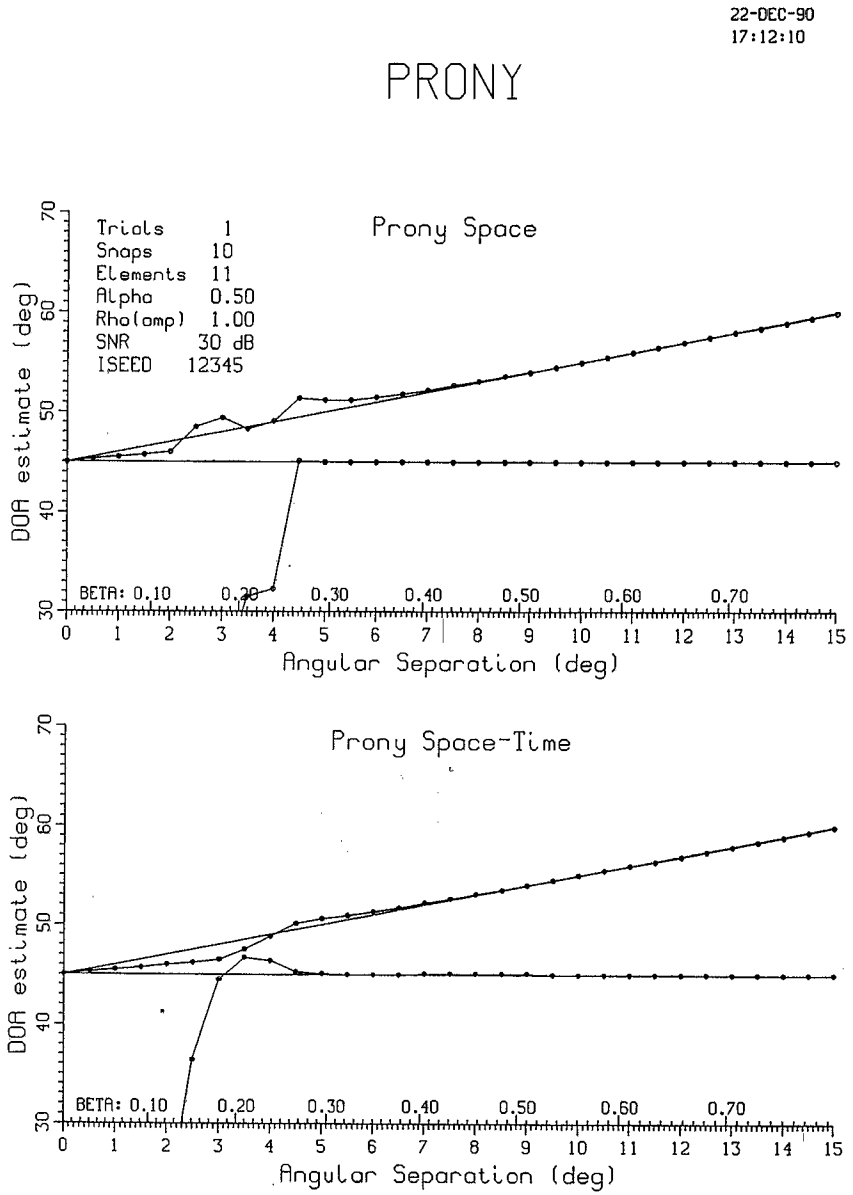


Fig. C3.5 - PRONY algorithm; perfect correlation; 30 dB SNR; 10 snapshots

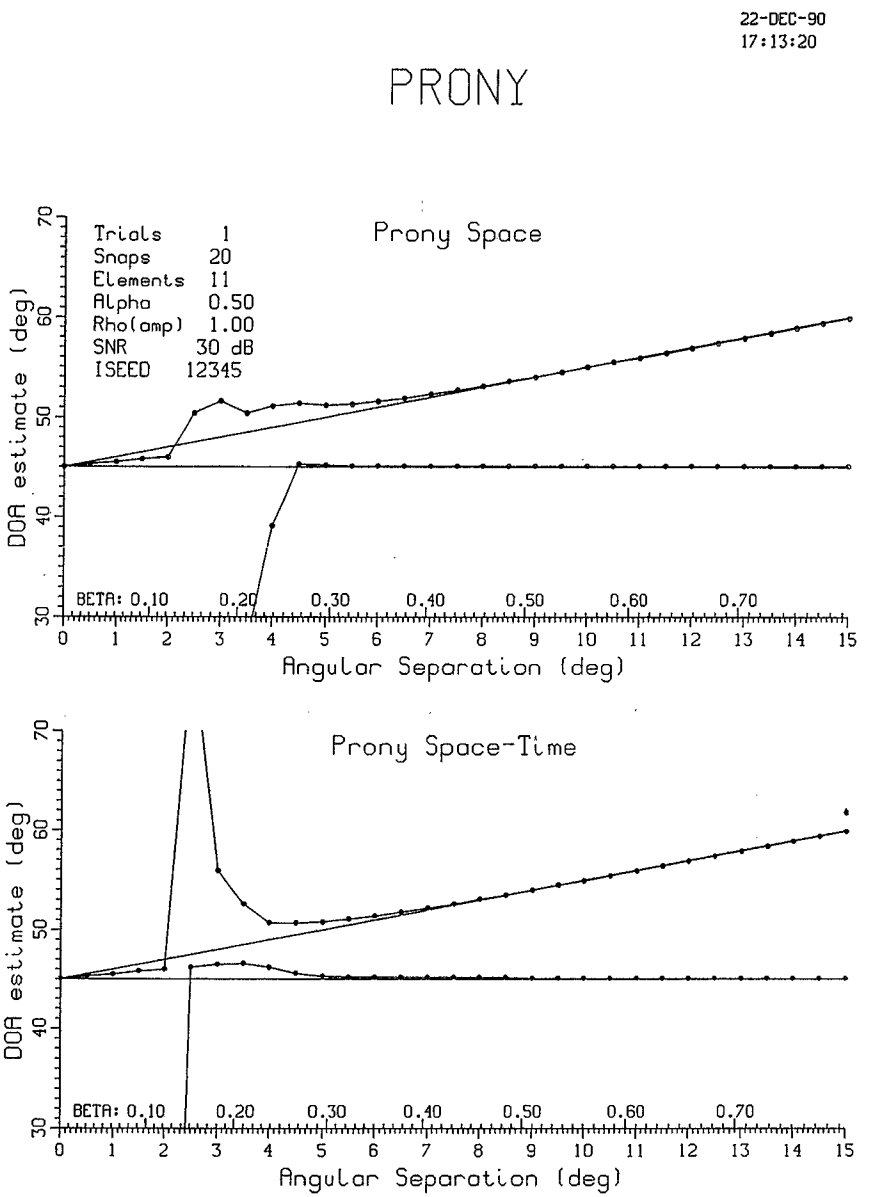


Fig. C3.6 - PRONY algorithm; perfect correlation; 30 dB SNR; 20 snapshots

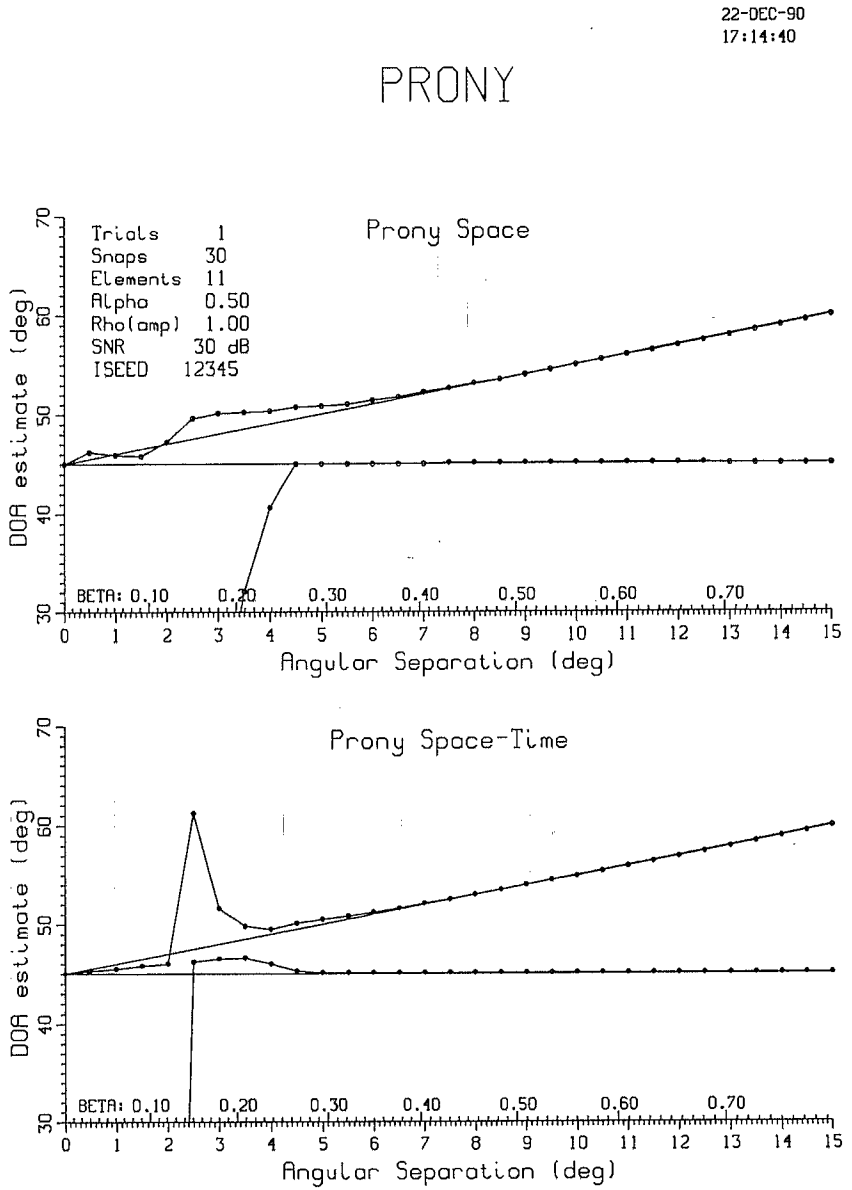


Fig. C3.7 - PRONY algorithm; perfect correlation; 30 dB SNR; 30 snapshots

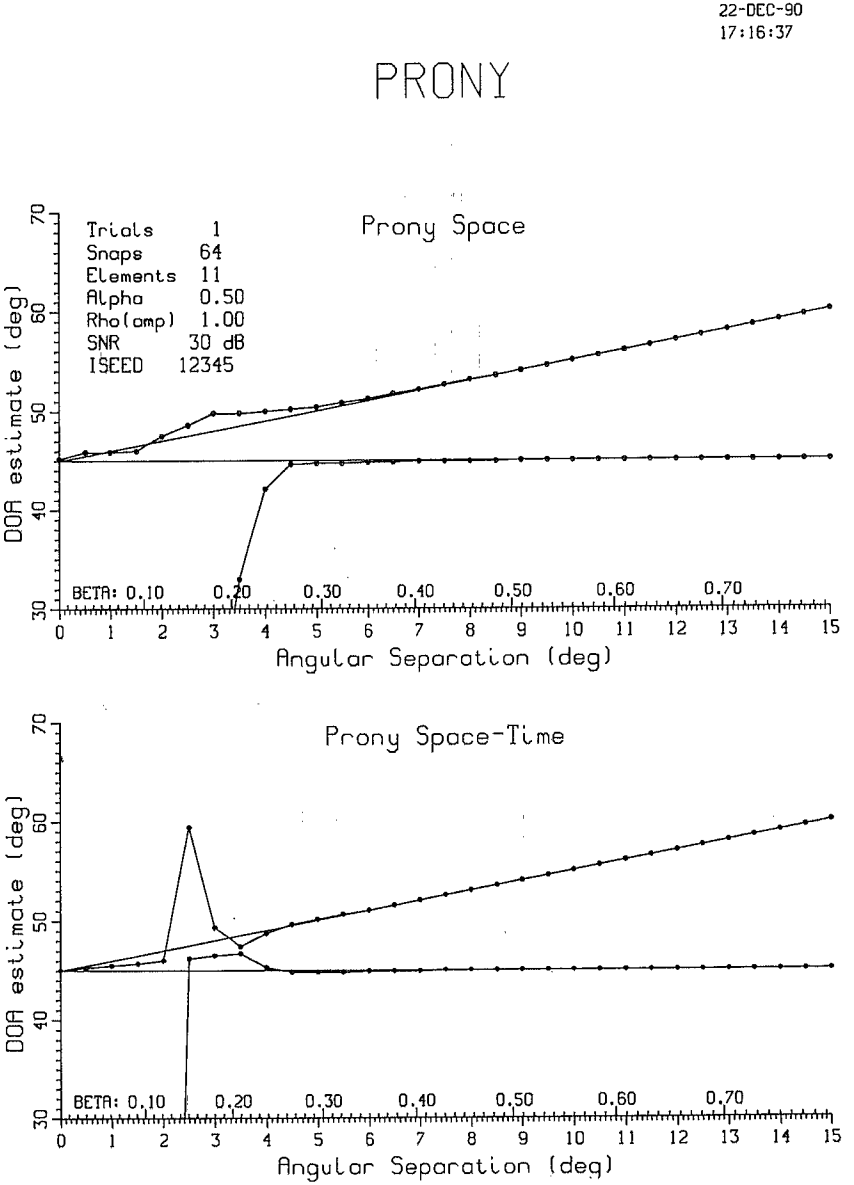


Fig. C3.8 - PRONY algorithm; perfect correlation; 30 dB SNR; 64 snapshots

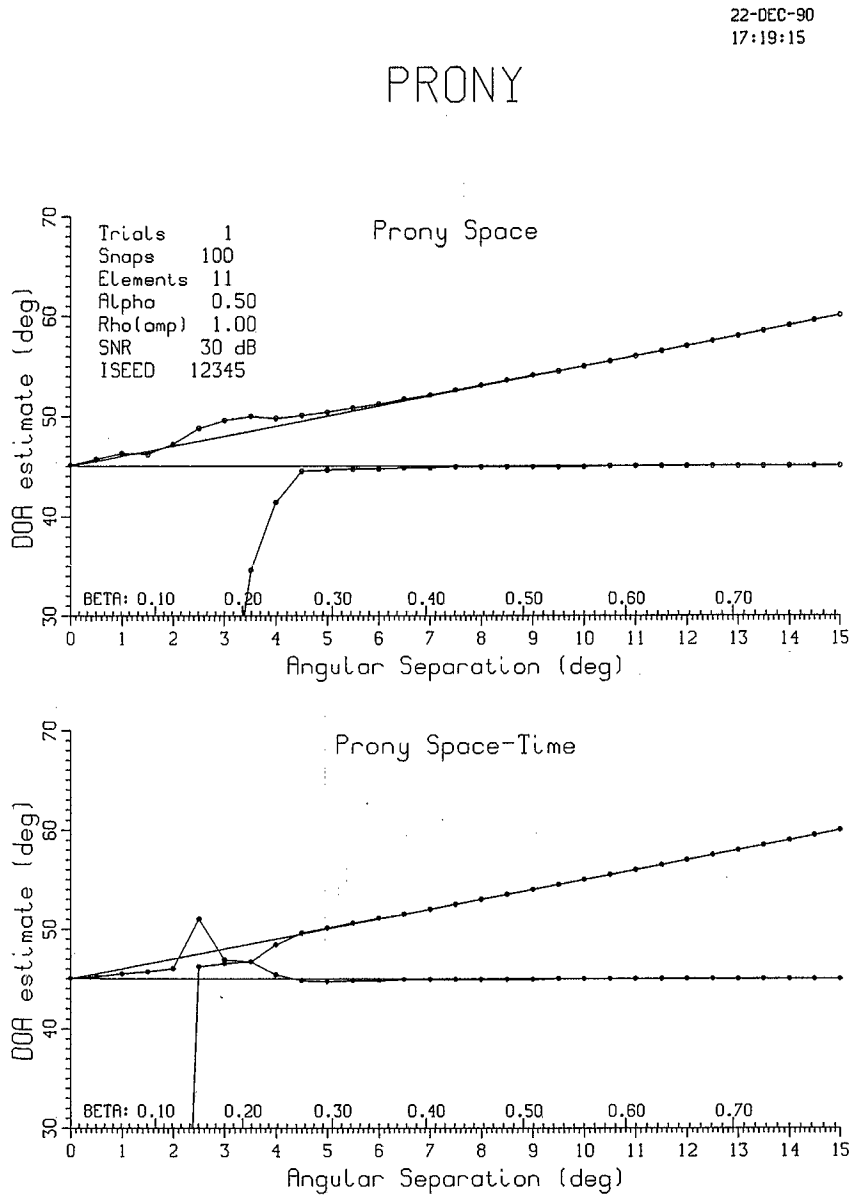


Fig. C3.9 - PRONY algorithm; perfect correlation; 30 dB SNR; 100 snapshots

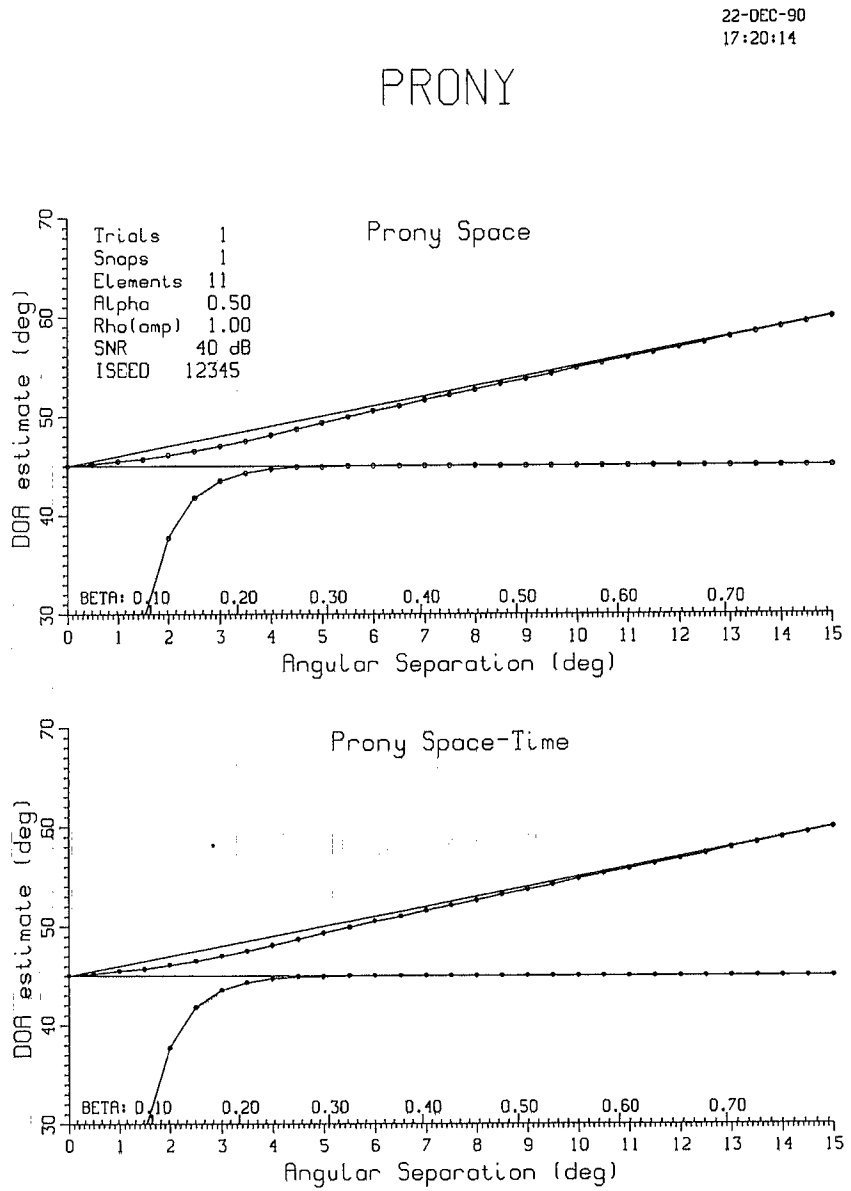


Fig. C4.1 - PRONY algorithm; perfect correlation; 40 dB SNR; 1 snapshot

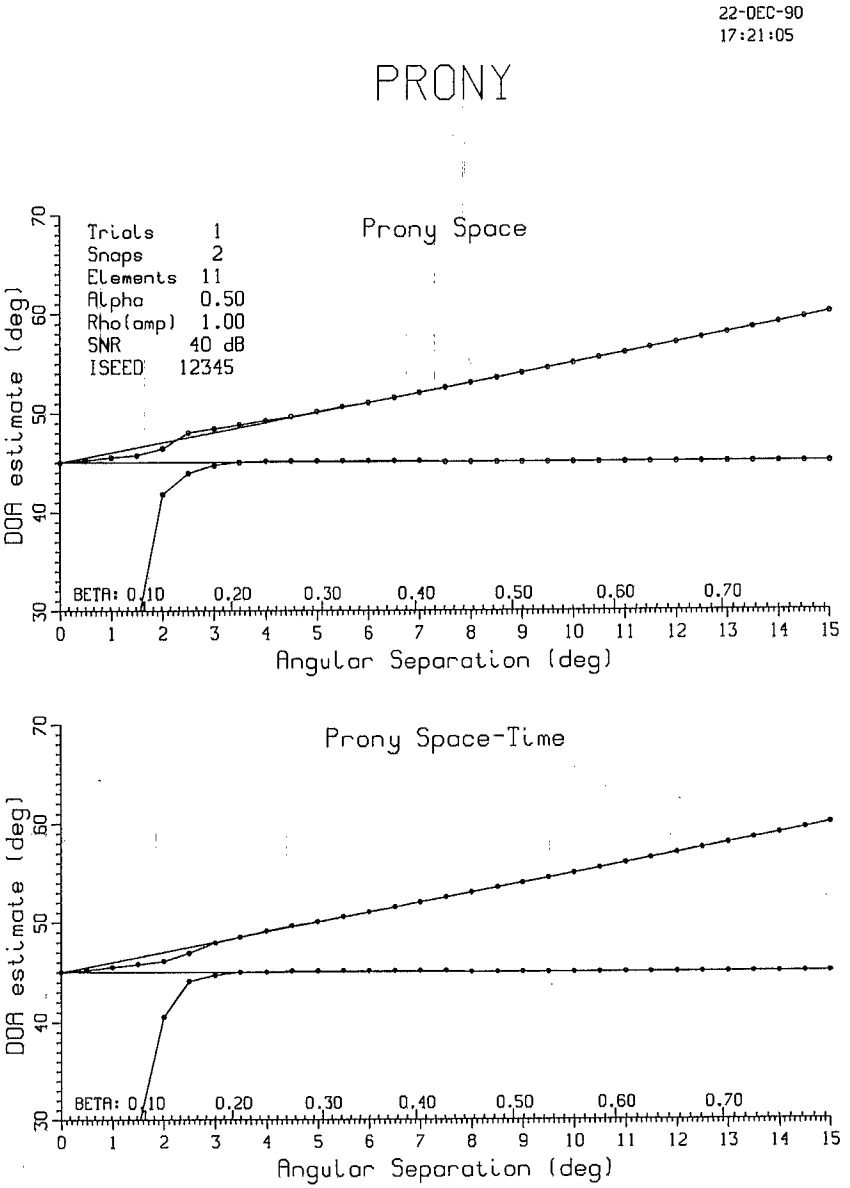


Fig. C4.2 - PRONY algorithm; perfect correlation; 40 dB SNR; 2 snapshots

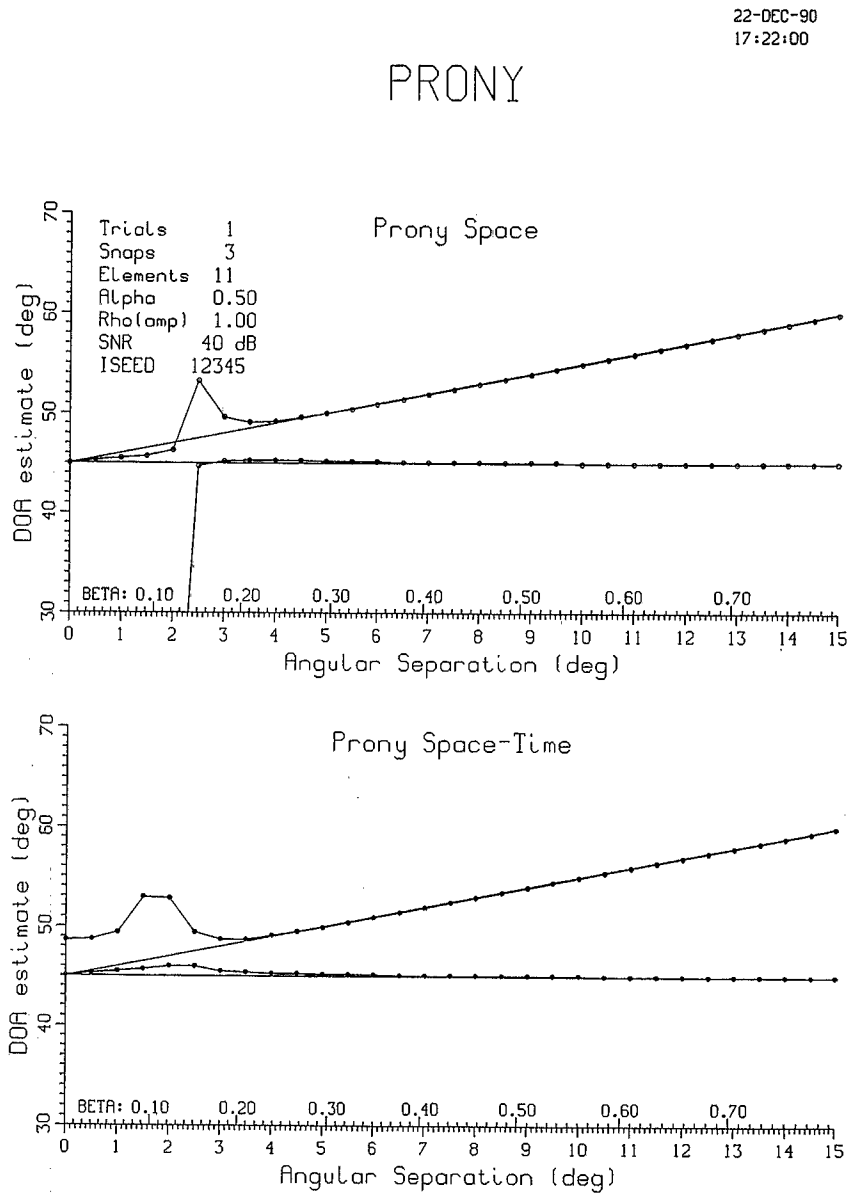


Fig. C4.3 - PRONY algorithm; perfect correlation; 40 dB SNR; 3 snapshots

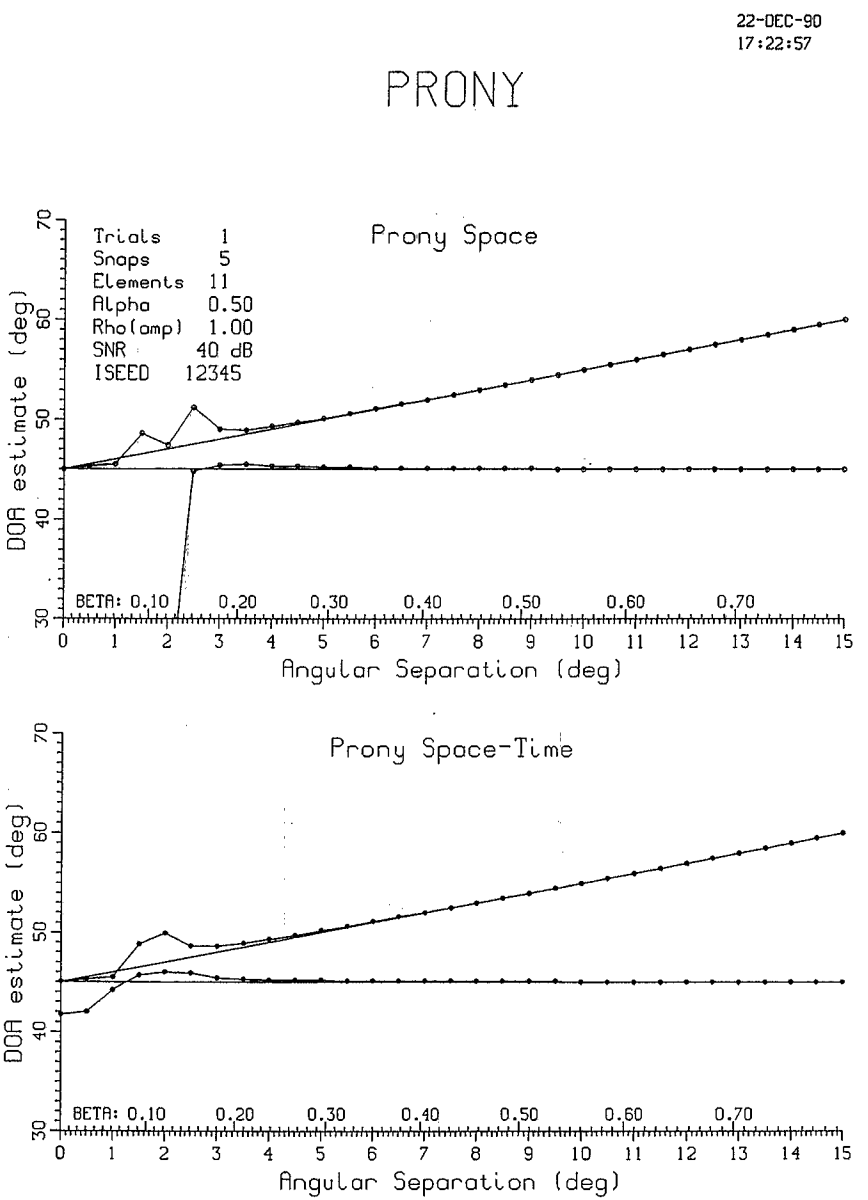


Fig. C4.4 - PRONY algorithm; perfect correlation; 40 dB SNR; 5 snapshots

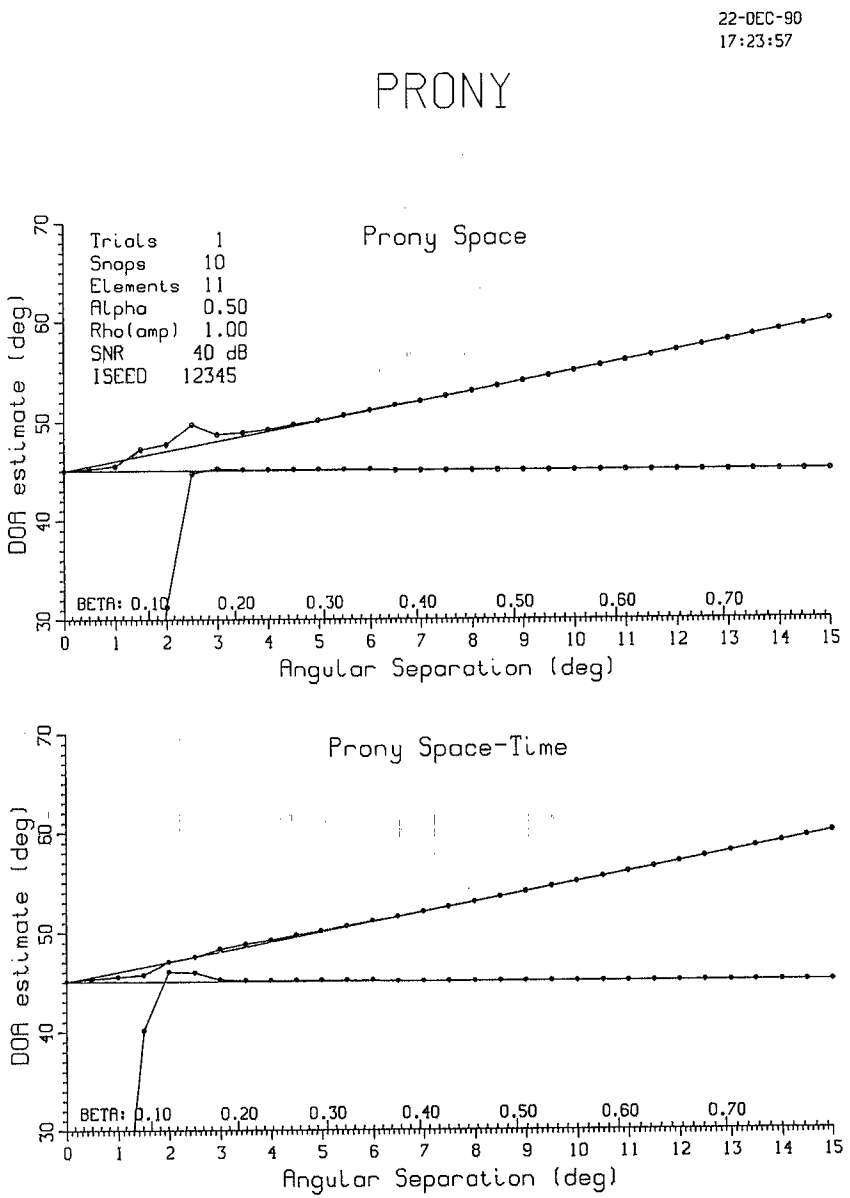


Fig. C4.5 - PRONY algorithm; perfect correlation; 40 dB SNR; 10 snapshots

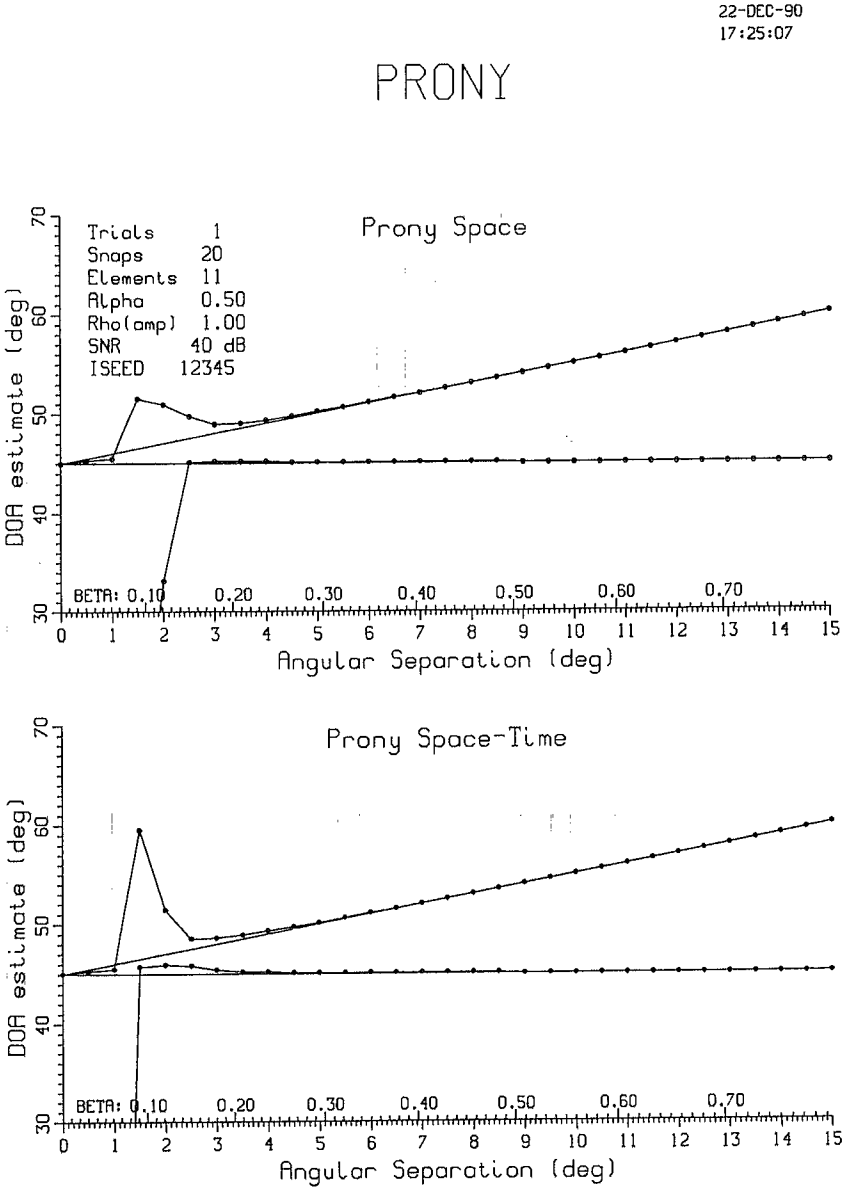


Fig. C4.6 - PRONY algorithm; perfect correlation; 40 dB SNR; 20 snapshots

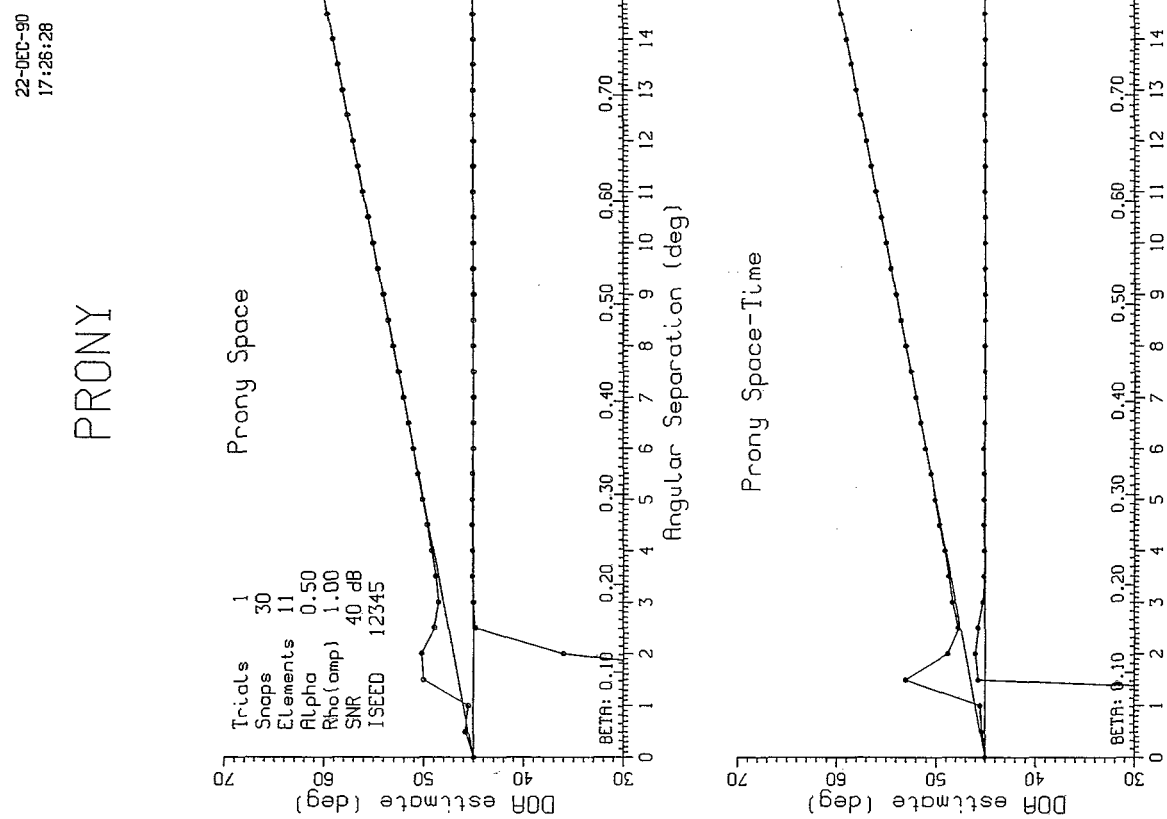


Fig. C4.7 - PRONY algorithm; perfect correlation; 40 dB SNR; 30 snapshots

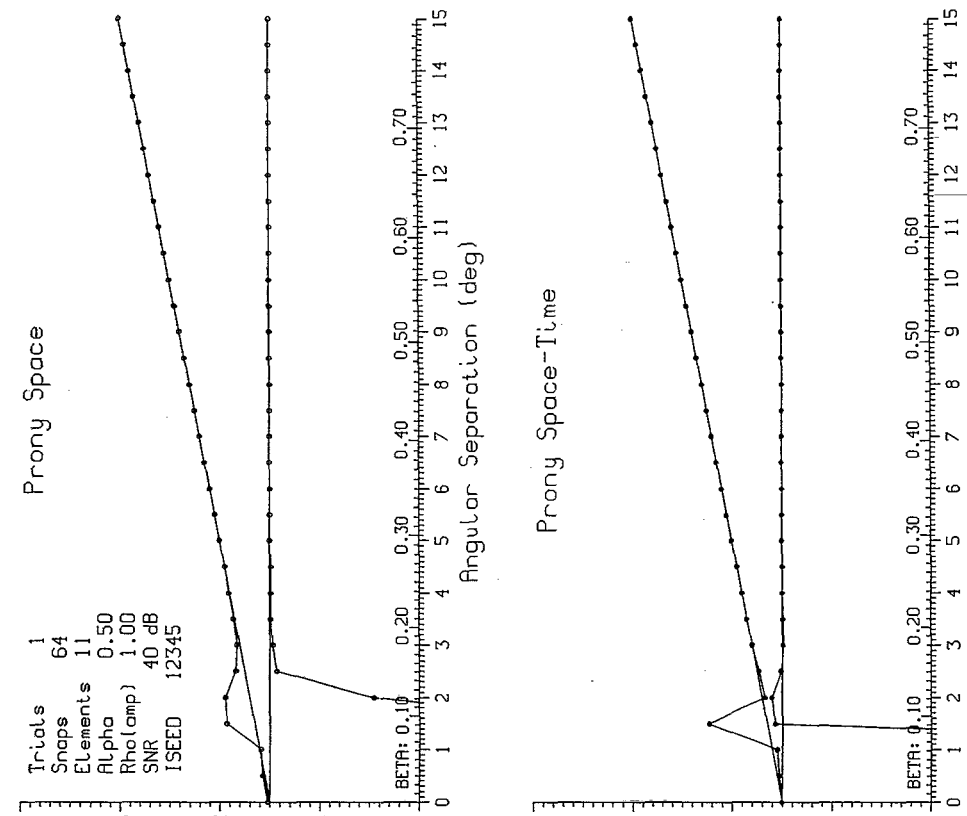


Fig. C4.8 - PRONY algorithm; perfect correlation; 40 dB SNR; 64 snapshots

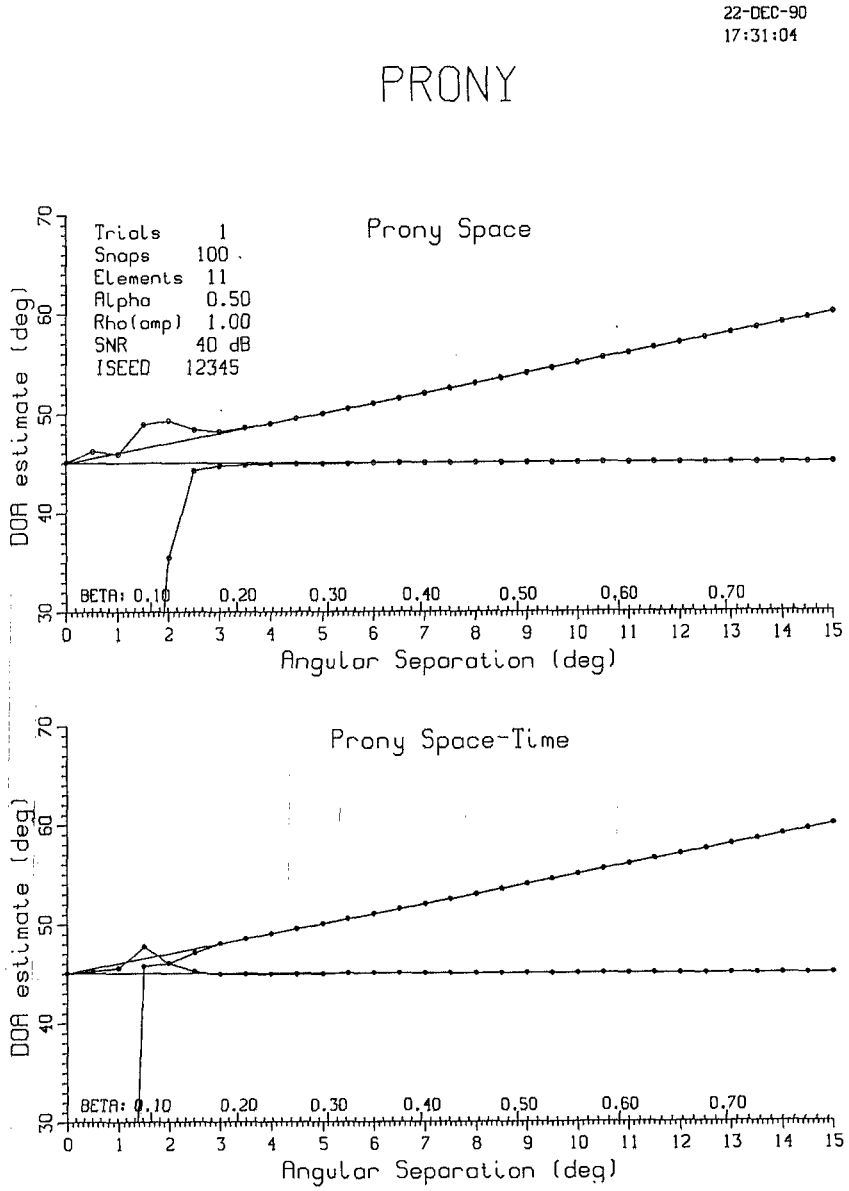


Fig. C4.9 - PRONY algorithm; perfect correlation; 40 dB SNR; 100 snapshots

Appendix A

ESPRIT ALGORITHM

The ESPRIT algorithm [A1] is an attempt to achieve performance as good as that of MUSIC, without having to generate the time- and memory-consuming array manifold (steering vectors). ESPRIT uses two or more arrays that are translated (or rotated) with respect to each other, and then takes advantage of the invariance of the signal subspace under translation to solve a generalized eigenvalue equation whose eigenvalues give the directions of arrival. (We must assume stationarity of the noise between the arrays, although we do not need to assume a specific structure for the noise component of the correlation matrix, e.g., diagonal, as we did with MUSIC.) In the following, we consider only the case of a translated array.

The signal at the elements of the shifted array is related to that at the elements of the unshifted array by a *phase change* [A2]:

$$\begin{aligned} A_1\, e^{-ik_x x} &\rightarrow A_1\, e^{-ik_x(x+\Delta)} = A_1\, e^{-ik_x x}\, e^{-ik_x \Delta} \quad (x_m = m\,d\,,\ \delta = \Delta / d) \\ A_1\, e^{-ik_1 m} &\rightarrow A_1\, e^{-ik_1(m+\delta)} = A_1\, e^{-ik_1 m}\, e^{-ik_1 \delta} \quad (k_1 = k_x d)\,. \end{aligned} \tag{A1}$$

(This is the projection of one plane wave over the elements; we take the sum of expressions like this to get the full signal. A_1 is the amplitude; k_x is the projected wavevector; x is the position along the array in real space; Δ is the shift of the second array in real space; m is the number of the element; d is the element spacing; k_1 is a dimensionless wavevector that gives the signal in "digital" space, i.e., we are now analyzing a set of discrete samples with normalized frequencies and dimensionless time interval equal to 1, etc.)

With the following steering vector

$$S_k = \begin{bmatrix} 1 \\ e^{ik} \\ e^{i2k} \\ . \\ . \\ . \\ e^{iMk} \end{bmatrix} \quad k = \frac{2\pi d}{\lambda} \sin(\theta) \quad S = [s_{k_1}\ s_{k_2} \cdots s_{k_L}]\,, \tag{A2}$$

we can write for the total signal across elements

$$\underline{y}\,(n) = \sum_{i=1}^{L} A_i(n)\, \underline{s}^*_{k_i} + \underline{n}(n)\,. \tag{A3}$$

We define the correlation matrix as in MUSIC

$$\mathbf{R} = E\,[\underline{y}(n)^*\,\underline{y}(n)^T] = \sum_{i,j=1}^{L} \underline{s}_{k_i}\, P_{ij}\, \underline{s}_{k_j}^H + \sigma_v^2\, I \tag{A4}$$

$$P_{ij} = E\,[\,A_i(n)^*\,A_j(n)] \qquad (1 \le i,j \le L) \tag{A5}$$

with the following restrictions on the noise (although the left-hand "white" noise condition is not necessary in the general case if the noise statistics are somehow known)

$$E[\nu_m\,(n)^*\,\nu_k\,(n)] = \sigma_v^2\,\delta_{mk} \qquad E[\nu_m\,(n)^*\,A_i\,(n)] = 0 \tag{A6}$$

$$\mathbf{R} = \mathbf{S}\,\mathbf{P}\,\mathbf{S}^H + \sigma_v^2\,I\,. \tag{A7}$$

The steering matrix **S** transforms as

$$\mathbf{S} \rightarrow \mathbf{S}\,\mathbf{D}_\delta \quad \mathbf{D}_\delta = \begin{bmatrix} e^{-ik_1\delta} & 0 & 0 & \cdots & 0 \\ 0 & e^{-ik_2\delta} & 0 & \cdots & 0 \\ . & . & . & \cdots & . \\ 0 & 0 & 0 & \cdots & e^{-ik_L\delta} \end{bmatrix}. \tag{A8}$$

This yields: for the signal at the unshifted and shifted arrays,

$$\underline{y}(n) = \mathbf{S}^*\,\mathbf{A}(n) + \underline{\nu}\,(n) \tag{A9}$$

$$\underline{y}_\delta(n) = \mathbf{S}^*\,\mathbf{D}_\delta^*\,\mathbf{A}(n) + \underline{\nu}_\delta(n)\,; \tag{A10}$$

for the unshifted and shifted correlation matrices,

$$\mathbf{R}_{yy} = E[\underline{y}(n)^*\,\underline{y}(n)^T] = \mathbf{S}\,\mathbf{P}\,\mathbf{S}^H + \sigma_v^2\,I \tag{A11}$$

and

$$\mathbf{R}_{y_\delta y_\delta} = E[\underline{y}_\delta(n)^*\,y_\delta(n)^T] = \mathbf{S}\,\mathbf{D}_\delta\,\mathbf{P}\,\mathbf{D}_\delta^H\,\mathbf{S}^H + \sigma_v^2\,I\,; \tag{A12}$$

and for the cross-correlation matrix between the unshifted and shifted arrays,

$$\mathbf{R}_{yy_\delta} = E\,[\underline{y}(n)^*\,\underline{y}_\delta(n)^T] = \mathbf{S}\,\mathbf{P}\,\mathbf{D}_\delta^H\,\mathbf{S}^H\,, \tag{A13}$$

$$E[\underline{\nu}_\delta(n)^*\,\underline{\nu}_\delta(n)^T] = E[\underline{\nu}(n)^*\,\underline{\nu}(n)^T] = \sigma_v^2\,I\ , \tag{A14}$$

and

$$E[\underline{y}(n)^*\,\underline{\nu}_\delta(n)^T] = 0. \tag{A15}$$

We can now define the so-called *matrix pencil*

$$\mathbf{C}(\lambda) = \mathbf{C} - \lambda\mathbf{C}_\delta\,, \tag{A16}$$

where

$$\mathbf{C} = \mathbf{R}_{yy} - \sigma_v^2 \mathbf{I} = \mathbf{S}\,\mathbf{P}\,\mathbf{S}^H \tag{A17}$$

and

$$\mathbf{C}_\delta = \mathbf{R}_{yy_\delta} = \mathbf{S}\,\mathbf{P}\,\mathbf{D}_\delta^H\,\mathbf{S}^H . \tag{A18}$$

These correlation matrices satisfy a so-called *generalized eigenvalue problem*

$$\det(\mathbf{C} - \lambda\mathbf{C}_\delta) = 0 \quad \mathbf{C}\,\underline{e} = \lambda\,\mathbf{C}_\delta\,\underline{e} \tag{A19}$$

whose L *nonzero* eigenvalues give the directions of arrival

$$\lambda_i = e^{ik_i\delta} \quad (i = 1,2, \ldots L) . \tag{A20}$$

This can be seen by substituting the expressions above for $\mathbf{C}$ and $\mathbf{C}_\delta$ in the eigenvalue equation. Premultiplying both sides by $\mathbf{S}^H$ and removing the common factor $(\mathbf{S}^H\mathbf{S})\mathbf{P}$, yields

$$\mathbf{S}^H\,\underline{e} = \lambda\,\mathbf{D}_\delta^H\,\mathbf{S}^H\,\underline{e} . \tag{A21}$$

Since $\mathbf{D}_\delta$ is diagonal, we have that $\mathbf{D}_\delta^{-1}$ is equal to $\mathbf{D}_\delta^H$, so that it is easy to transfer $\mathbf{D}_\delta$ to the other side of the equation. If we then define $\mathbf{S}^H\underline{e}$ as an eigenvector of $\mathbf{D}_\delta$, we get (by using the diagonality of $\mathbf{D}_\delta$) the eigenvalue relation above, i.e., $\lambda_i = \mathbf{D}_\delta(i,i)$.

In practice, we estimate the correlation matrices from the data, as with MUSIC. In particular, the cross-correlation matrix is found from

$$\hat{\mathbf{C}}_\delta = \hat{\mathbf{R}}_{yy_\delta} = \frac{1}{N}\sum_{n=0}^{N-1} \underline{y}(n)^*\,\underline{y}_\delta(n)^T . \tag{A22}$$

Figures D1.1 through D2.4 are some examples of results from my ESPRIT program. The shift between the two arrays is indicated by Delta(d), which indicates the fraction of an element spacing that separates the arrays. In practice, the array is intended to consist of matched pairs of elements rather than widely spaced arrays. As before, the element spacing is $\lambda/2$. We consider only the case of perfect decorrelation, although the method should work for perfect correlation as well (with suitable modifications).

Figures D1.1 through D1.4 compare MUSIC and ESPRIT in the case of very low noise (SNR = 99 dB). Both algorithms give exact results, suggesting that they both work properly. (It should be noted, though, that for the noiseless case, the generalized eigenvalue algorithm failed to converge for certain values of delta, such as 0.2, 0.5, and 1.0. This is probably due to some kind of singularity or ill-posedness in the matrices. This did not occur in the noisy case.) Figures D2.1 through D2.4 present a comparison of the two in 30 dB noise for the same array separations, which seems to indicate that ESPRIT works even better than does MUSIC.

REFERENCES

A1. R. Roy and T. Kailath, "ESPRIT - Estimation of Signal Parameters Via Rotational Invariance Techniques," *IEEE Trans. Acous., Speech, Sig. Proc.*, **37**(7), 984 (1989).

A2. S. J. Orphanidis, *Optimum Signal Processing* (Macmillan, New York, N.Y., 1988).

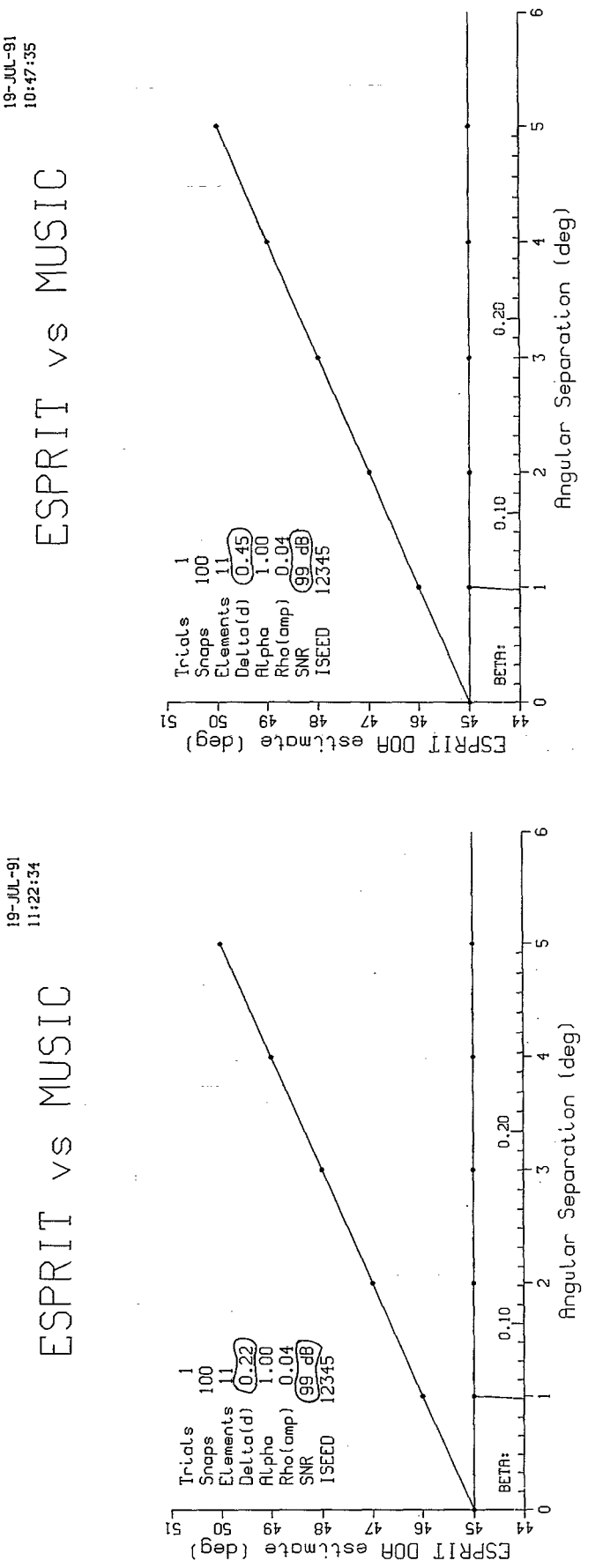


Fig. D1.1 - ESPRIT algorithm; perfect decorrelation; 99 dB SNR; delta = 0.22

Fig. D1.2 - ESPRIT algorithm; perfect decorrelation; 99 dB SNR; delta = 0.45

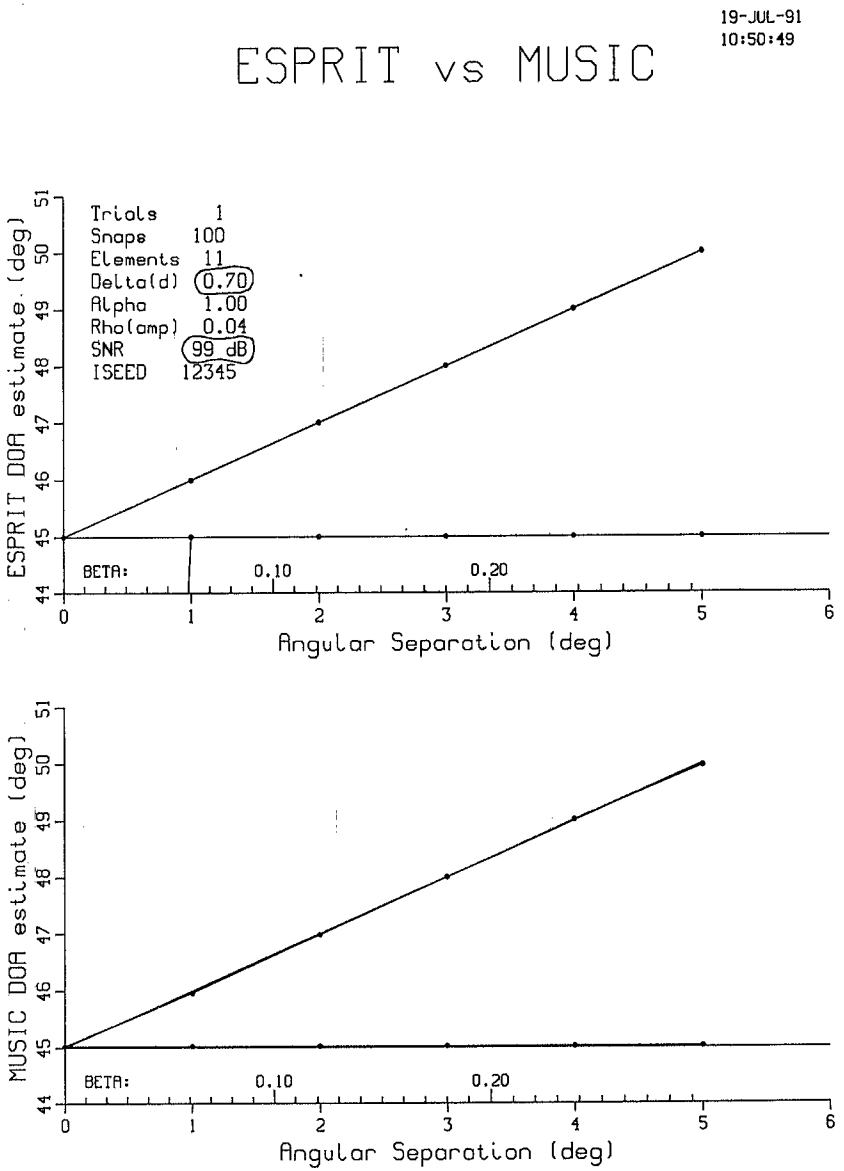


Fig. D1.3 - ESPRIT algorithm; perfect decorrelation; 99 dB SNR; delta = 0.70

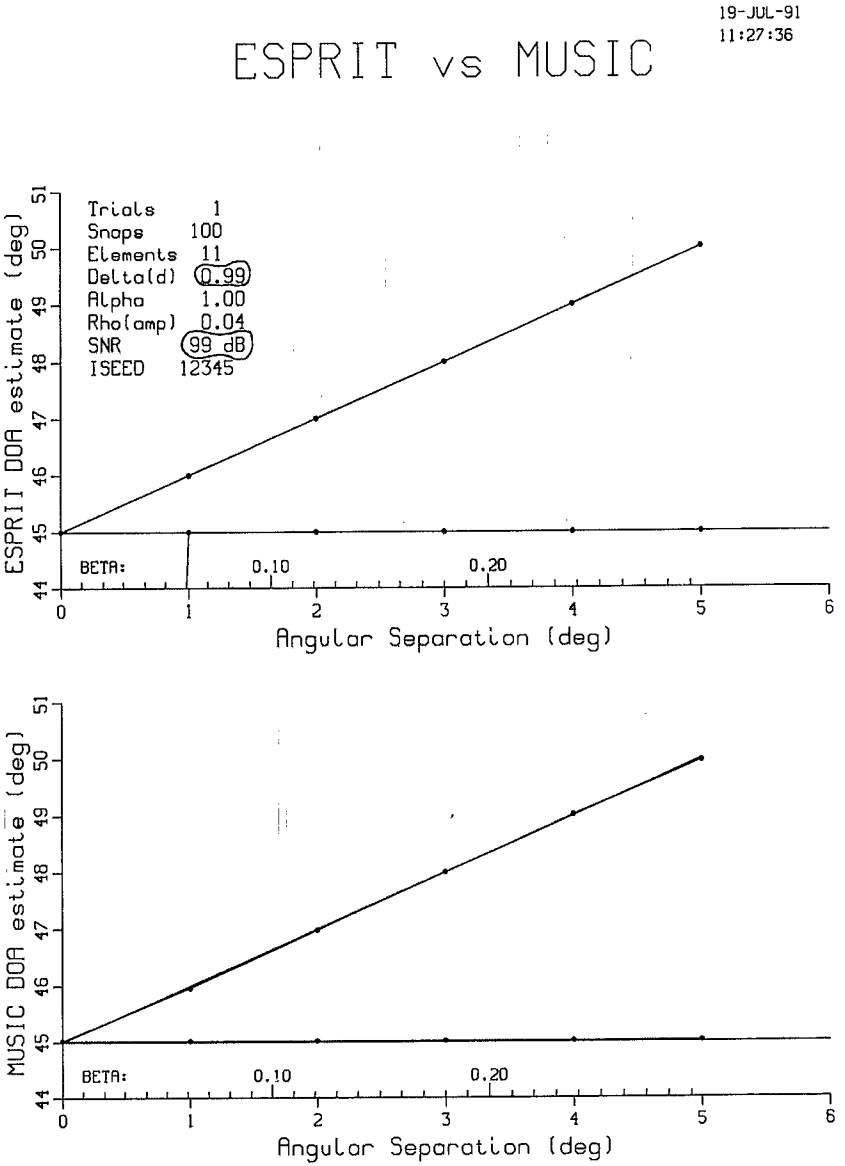


Fig. D1.4 - ESPRIT algorithm; perfect decorrelation; 99 dB SNR; delta = 0.99

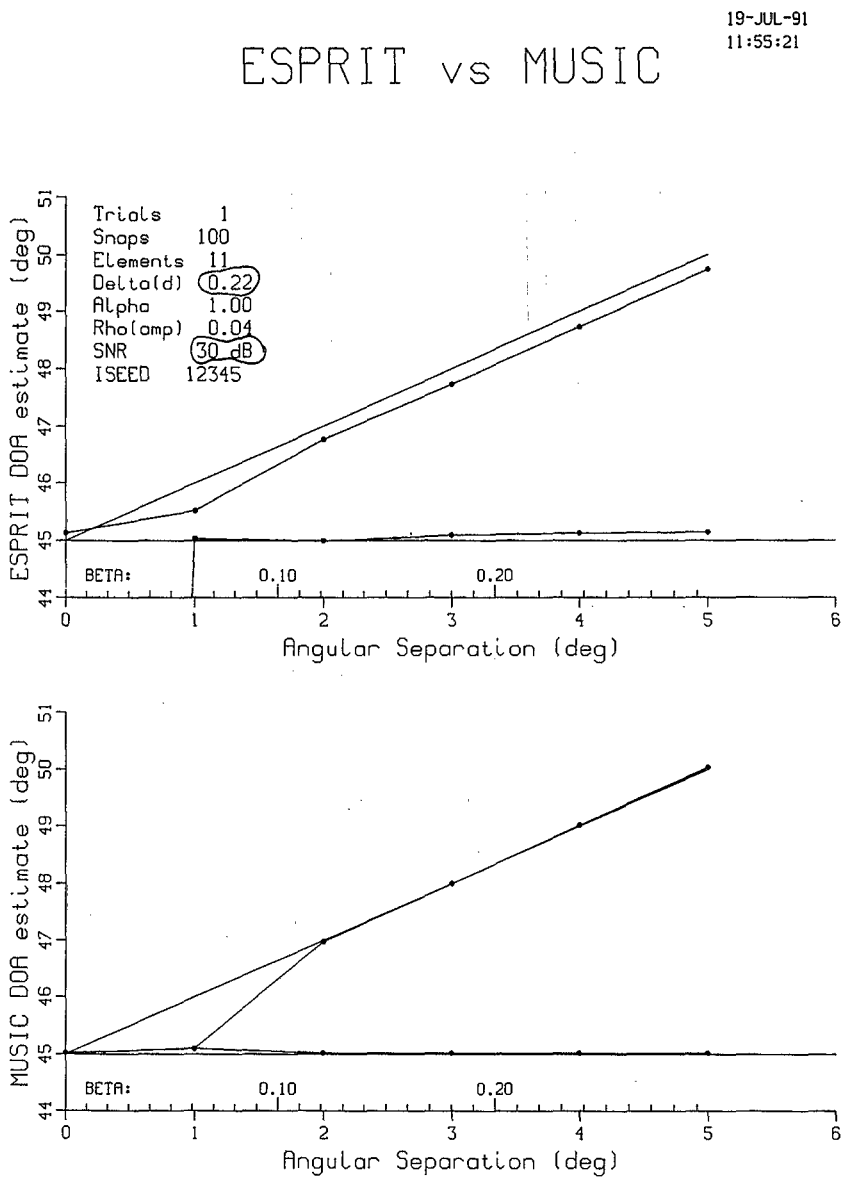


Fig. D2.1 - ESPRIT algorithm; perfect decorrelation; 30 dB SNR; delta = 0.22

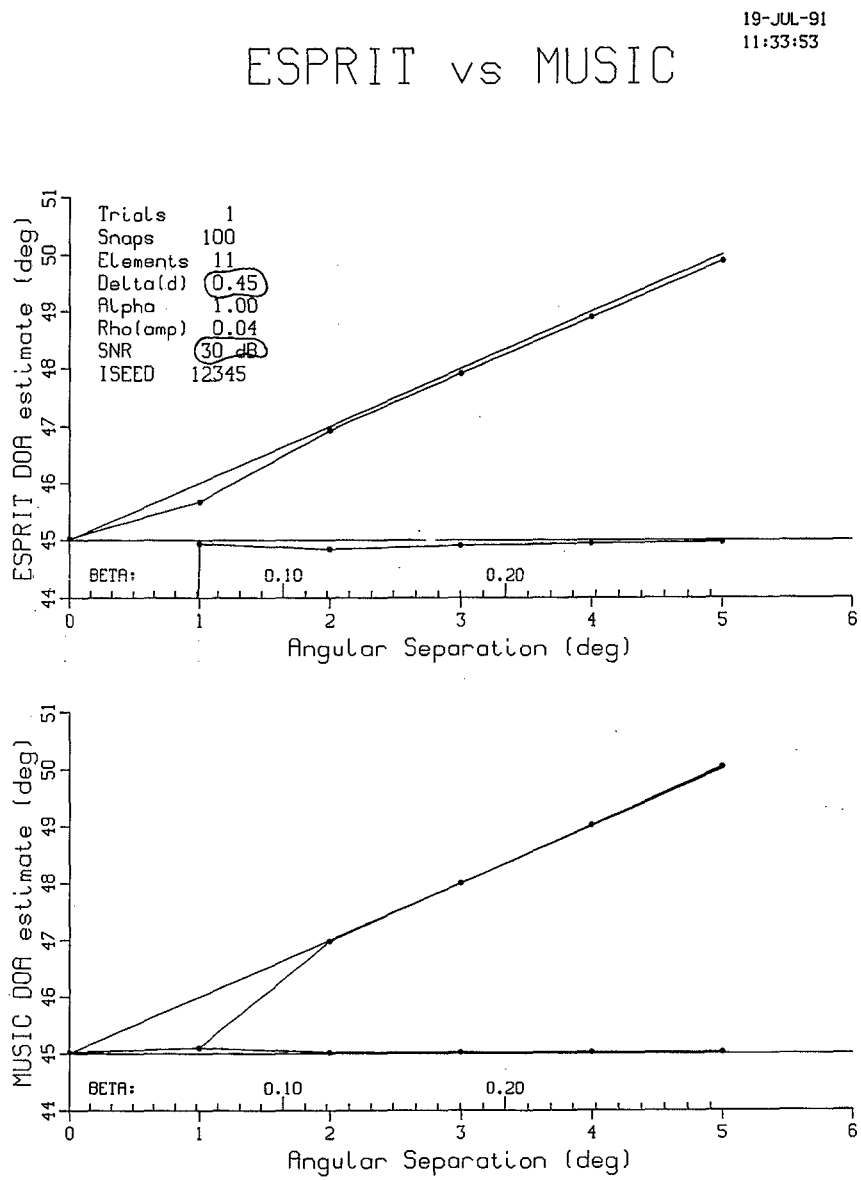


Fig. D2.2 - ESPRIT algorithm; perfect decorrelation; 30 dB SNR; delta = 0.45

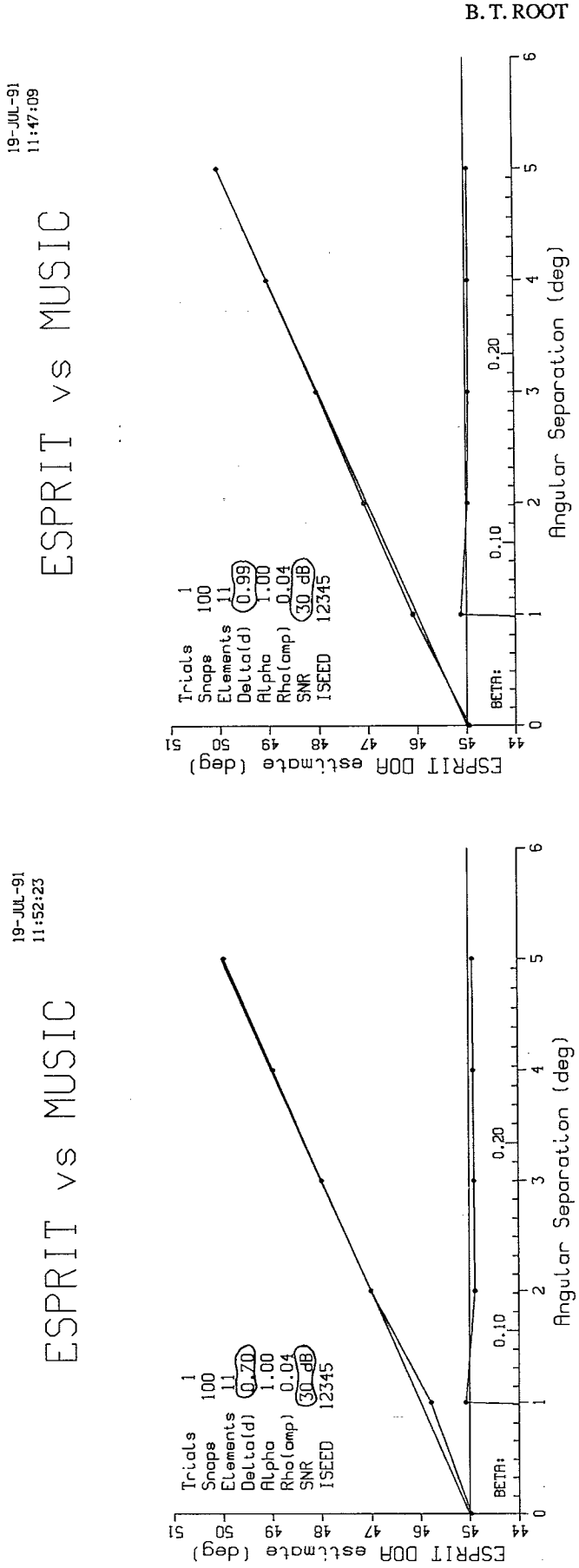


Fig. D2.3 - ESPRIT algorithm; perfect decorrelation; 30 dB SNR; delta = 0.70

Fig. D2.4 - ESPRIT algorithm; perfect decorrelation; 30 dB SNR; delta = 0.99

Appendix B

SPATIALLY AVERAGED MUSIC

Spatially averaged MUSIC is a variation of MUSIC that works in the case of perfect correlation between the signals. Essentially, it divides the array into subarrays, calculates a correlation matrix for each subarray in the usual way, and then averages these together. It can be shown through straightforward substitution and algebraic manipulation that the singularity in the correlation matrix is removed by this averaging process* . The different subarrays provide a slightly different "perspective" on the incoming signals, and this extra degree of freedom overcomes the lack of full rank in the unaveraged correlation matrix. The price we must pay is that we are using smaller arrays, and it is not immediately clear that the decrease in resolution is acceptable. This appendix presents some results which show that we get performance in the correlated case that rivals that of the Prony method presented in the main body of this report.

Table B1 presents a comparison of spatially averaged MUSIC and Prony. There are five elements in each subarray of the spatially averaged case. All other quantities are equivalent to the analysis presented in the report. "N" refers to the number of snapshots. The numbers inside the table are the angular thresholds for resolution, i.e., the angular separation in degrees at which the two perfectly correlated signals were first resolved. The numbers above the diagonals are for the averaged MUSIC case and below the diagonals are for the Prony case as presented in this report. Clearly, the performance is comparable.

Table B1—Comparison of Spatially Averaged MUSIC and Prony

	N=2	N=10	N=100
SNR = 10 dB	10 (MUSIC) / 11 (PRONY)	10 / 10	7 / 10
SNR = 20 dB	7.5 / 7	7 / 7	4.5 / 7
SNR = 30 dB	5.5 / 4.5	4.5 / 4.5	3 / 4.5

*S.J. Orphanidis, *Optimum Signal Processing*, Macmillan, New York, N.Y., 1988.

Appendix C

SOME RECENT RESEARCH IN MUSIC

The following remarks survey some of the recent literature on research into the MUSIC algorithm.

MUSIC is highly sensitive to system errors. Excellent calibration is difficult to accomplish, and even when possible, the system parameters will drift with time. Reference C1 attempts to quantify this sensitivity by a first-order analysis, assuming a perfectly known covariance matrix. The basic conclusion is that increasing array aperture will reduce the sensitivity, so that there is a tradeoff between the costs of calibration and of increased aperture.

A beamforming preprocessor may increase the performance of MUSIC and of other high-resolution algorithms. Benefits include reduced computation, improved performance in the presence of spatially colored noise, and increased resolution. Reference C2 provides an in-depth, quantitative analysis for closely spaced, uncorrelated targets.

Root-MUSIC is a variation of MUSIC that has superior performance in the case of a linear, equispaced array. This method determines the direction-of-arrivals by solving a polynomial formed from the noise subspace, rather than scanning numerically over all DOAs to find the orthogonality between signal and noise subspaces described in this report. Reference C3 explains and quantifies Root-MUSIC with a perturbation analysis on the roots of the polynomial.

In Reference C4 an asymptotic analysis is presented for a generalized version of the smoothing scheme for coherent sources discussed in Appendix B of this report. This generalization is called *forward-backward* smoothing as compared to the forward only smoothing in Appendix B. Forward only smoothing requires 2K elements to detect K coherent DOAs, whereas forward-backward smoothing requires only [3K/2] elements. In Pillai's analysis, a uniform array is assumed, which is a restriction on the arbitrary array geometry of which MUSIC is capable. A key result is that about $(1/M\omega_d)^2 - 1$ more snapshots are required to resolve two equipowered closely spaced coherent plane waves than are required for uncorrelated waves, where M is the number of elements and ω_d is the angular separation. Statistics for the unsmoothed and forward-only smoothed cases result as special cases of Pillai's general analysis.

Reference C5 provides analytic results for the asymptotic efficiency of MUSIC at high SNR with multiple, uncorrelated targets. That is, when the number of snapshots approaches infinity, the variances of the DOAs approach the Cramer-Rao lower bound.

Minimum-Norm is another leading contender besides MUSIC for array high resolution. An asymptotic statistical analysis of these two algorithms is provided in Reference C6. Results suggest that Min-Norm has a resolution threshold at a lower SNR than MUSIC.

Reference C7 presents an analysis of the spatially averaged method for correlated signals discussed in Appendix B.

REFERENCES

C1. B. Friedlander, "A Sensitivity Analysis of the MUSIC Algorithm," *IEEE Trans. Acous., Speech, Sig. Proc.*, **38**(10), 1740 (1990).

C2. H. B. Lee and M. S. Wengrovitz, "Resolution Threshold of Beamspace MUSIC for Two Closely Spaced Emitters," *IEEE Trans. Acous., Speech, Sig. Proc.*, **38**(9), 1545 (1990).

C3. B. D. Rao and K. V. S. Hari, "Performance Analysis of Root-MUSIC," *IEEE Trans. Acous., Speech, Sig. Proc.*, **37**(12), 1939 (1989).

C4. S. U. Pillai and B. H. Kwon, "Performance Analysis of MUSIC-type High Resolution Estimators for Direction Finding in Correlated and Coherent Scenes," *IEEE Trans. Acous., Speech, Sig. Proc.*, **37**(8), 1176 (1989).

C5. B. Porat and B. Friedlander, "Analysis of the Asymptotic Relative Efficiency of the MUSIC Algorithm," *IEEE Trans. Acous., Speech, Sig. Proc.*, **36**(4), 532 (1988).

C6. M. Kaveh, "The Statistical Performance of the MUSIC and Minimum-Norm Algorithms in Resolving Plane Waves in Noise," *IEEE Trans. Acous., Speech, Sig. Proc.*, **ASSP-34**(2), 1331 (1986).

C7. T-J Shan, M.Wax, and T. Kailath, "On Spatial Smoothing for Direction-of-Arrival Estimation of Coherent Signals," *IEEE Trans. Acous., Speech, Sig. Proc.*, **ASSP-33**(4), 806 (1985).